

Oracle® Real-Time Decisions Base Application

Installation and Reference Guide

Version 2.2.1

E12182-01

May 2008

Copyright © 2008, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software--Restricted Rights (June 1987). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

Contents

Preface	vii
Audience	vii
Documentation Accessibility	vii
Related Documents	viii
Conventions	viii

Part I Getting Started

1 Installation of Oracle RTD Base Application Modules

1.1 Installation Steps	1-1
------------------------------	-----

2 Overview of Oracle Real-Time Decisions

2.1 Introduction to the Oracle RTD Decision Process.....	2-1
2.2 Integration Points Between External Applications and Oracle RTD	2-4
2.3 Overview of Oracle RTD Integration with External Applications	2-4
2.4 The Oracle RTD Decision Process	2-6
2.5 More About the Oracle RTD Decision Process Elements.....	2-7
2.5.1 Performance Goals.....	2-7
2.5.2 Decisions	2-8
2.5.3 Choice Groups and Choices	2-8
2.5.4 Eligibility Rules	2-9
2.5.5 Filtering Rules	2-9
2.5.6 Scoring Rules	2-9
2.5.7 Models	2-9
2.5.7.1 Model Usage Summary	2-10
2.5.7.2 Choice Models.....	2-10
2.5.7.3 Choice Event Models	2-10
2.6 General Oracle RTD Elements and Features.....	2-11
2.6.1 Data Sources	2-11
2.6.2 Entities.....	2-11
2.6.3 Functions.....	2-12
2.7 Oracle RTD Inline Services	2-12
2.8 Introduction to Oracle RTD Decision Studio	2-13
2.9 Analytic Reports and the Oracle RTD Decision Center	2-14

Part II Base E-Commerce Inline Service

3 Base E-Commerce Inline Service Elements

3.1	Introduction to the Base E-Commerce Inline Service	3-1
3.2	Entity Logical Model	3-2
3.3	Choices Groups and Choices.....	3-2
3.3.1	Offer-Oriented Choices	3-3
3.3.2	Interaction-Oriented Choices	3-3
3.3.3	Informants Usage with Offer-Oriented and Transaction-Oriented Choices.....	3-5
3.4	Inline Service Advisors	3-6
3.4.1	Decisioning Advisors	3-6
3.4.1.1	Get Upsell Offers	3-6
3.4.1.2	Get Cross Sell Offers	3-9
3.4.1.3	Get Promotions	3-11
3.4.1.4	Get Advertisements.....	3-13
3.4.1.5	Get Offers.....	3-16
3.4.2	Analysis Advisors.....	3-18
3.4.2.1	Get Abandonment Propensity	3-18
3.4.2.2	Get Likely Web Action.....	3-19
3.4.2.3	Get Likely Web Duration	3-21
3.4.2.4	Get Web Support Types.....	3-23
3.5	Inline Service Informants.....	3-24
3.5.1	Initiate Session.....	3-25
3.5.2	Identify Customer.....	3-25
3.5.3	Page Turn.....	3-26
3.5.4	Performed Search.....	3-26
3.5.5	Customer Action.....	3-27
3.5.6	Added To Cart.....	3-27
3.5.7	Offer Response	3-28
3.5.8	Web Support Feedback	3-28
3.5.9	Close Session	3-29
3.6	Inline Service Entities	3-30
3.6.1	Session Entity	3-32
3.6.2	Ad Entity	3-33
3.6.3	Ad List Entity	3-33
3.6.4	Agent Interaction Entity	3-33
3.6.5	Campaign Entity	3-34
3.6.6	Campaign History Entity.....	3-34
3.6.7	Campaign Item Entity	3-35
3.6.8	Cart Item Entity.....	3-35
3.6.9	Clicked Ad Entity	3-35
3.6.10	Clicked Promotion Entity	3-36
3.6.11	Cross Sell Product List Entity	3-36
3.6.12	Current Web Interaction Entity	3-36
3.6.13	Customer Entity	3-37
3.6.14	Interaction History Entity.....	3-38
3.6.15	Organization Entity	3-40

3.6.16	Person Entity	3-40
3.6.17	Product Entity	3-41
3.6.18	Promoted Item Entity	3-41
3.6.19	Promotion Entity	3-41
3.6.20	Promotion List Entity	3-42
3.6.21	Purchased Item Entity	3-42
3.6.22	Purchase History Entity	3-42
3.6.23	Rank Offers Entity	3-43
3.6.24	Search Entity	3-43
3.6.25	Up Sell Product List Entity	3-44
3.6.26	Web Action Entity	3-44
3.6.27	Web Interaction Entity	3-44
3.6.27.1	Derivation of Web Interaction Attributes in Referencing Entities	3-45
3.6.28	Web Page Entity	3-46
3.7	Inline Service Functions	3-46

4 Configuring the Base E-Commerce Inline Service

4.1	Aligning Your Business Flow with Oracle RTD.....	4-1
4.2	Integrating Oracle RTD with the Customer Front End	4-2
4.3	Mapping Entity Attributes to Customer Data Sources	4-2
4.4	Adding Additional Logic as Necessary.....	4-2

Preface

This document describes the features of the Oracle Real-Time Decisions (Oracle RTD) Base Application.

Audience

This document is intended for the following Oracle RTD users:

- Technical users configuring Inline Services using Decision Studio
- Business users of Decision Center
- Administrators

Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at <http://www.oracle.com/accessibility/>.

Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

TTY Access to Oracle Support Services

Oracle provides dedicated Text Telephone (TTY) access to Oracle Support Services within the United States of America 24 hours a day, 7 days a week. For TTY support, call 800.446.2398. Outside the United States, call +1.407.458.2479.

Related Documents

For more information, see the following documents in the Oracle Real-Time Decisions platform Version 2.2.1 documentation set:

- *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*
- *Oracle Real-Time Decisions Decision Studio Reference Guide*
- *Oracle Real-Time Decisions Decision Center User Guide*
- *Oracle Real-Time Decisions Release Notes*
- *Oracle Real-Time Decisions New Features Guide Version 2.2.1.*

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Part I

Getting Started

Part I helps you get started with Oracle RTD Base Application, and contains the following chapters:

- [Chapter 1, "Installation of Oracle RTD Base Application Modules"](#)
- [Chapter 2, "Overview of Oracle Real-Time Decisions"](#)

Installation of Oracle RTD Base Application Modules

This chapter describes how to install an Oracle RTD Base Application module.

As prerequisites, you must have installed one of the Oracle RTD supported J2EE application servers, and you must have successfully deployed the Oracle RTD server on to this application server. Before starting the installation, verify that Oracle RTD is running and that the server logs do not contain any errors.

For information on installing the Oracle RTD Server, please refer to *Oracle Real-Time Decisions Installation and Administration of Oracle RTD*, which is available with the Oracle RTD platform software.

1.1 Installation Steps

Setting up an Oracle RTD Base Application module involves importing the Inline Service for the module into Oracle RTD Studio, and then configuring it to align with the customer's data sources and business applications.

To set up the module Inline Service, you must import the module Inline Service project into Oracle RTD Studio, as follows:

1. Create a directory for your Inline Services, for example, `C:\RTD_ILS`.

This directory will be referred to as `RTD_ILS_HOME` in this documentation.

2. Locate the zip file for the Base Application module that you want to install.

The zip file is located in the following directory:

```
\software\Oracle Real-Time Decisions Base Application\  
<module_name>
```

3. Unzip the Base Application module zip file, for example `RTD_Base_ECommerce.zip`, into `RTD_ILS_HOME`.

Unzipping the file creates a directory under `RTD_ILS_HOME`.

This sub-directory is referred to from now on as `MODULE_ILS_FOLDER`.

4. In Oracle RTD Studio, select File > Import.
5. From the Import window, select Existing Projects into Workspace.
6. If not already visible in the Projects area of the Import Projects dialog box, click the Browse button, and locate `RTD_ILS_HOME\MODULE_ILS_FOLDER`. For example, `C:\RTD_ILS\RTD_Base_ECommerce`.
7. Select the directory `MODULE_ILS_FOLDER` in the file dialog box and click Ok.

8. In the Projects area of the Import Projects dialog box, select the project name for the Base Application module.
9. Click the Finish button to load the project into Oracle RTD Studio.

The Base Application module project appears in the Inline Service Explorer view of Oracle RTD Studio.

You can now configure the Inline Service according to your business process requirements. For more information, see [Chapter 4, "Configuring the Base E-Commerce Inline Service."](#)

Overview of Oracle Real-Time Decisions

Oracle Real-Time Decisions (Oracle RTD) enables you to develop adaptive enterprise software solutions. These solutions are adaptive because they use rules and predictive models to continuously learn from business process transactions as those transactions are executing. By continuously learning in real time, the adaptive solutions that you develop can optimize the outcome of each transaction and of the associated business process.

This chapter presents an overview of Oracle RTD, and of the Oracle RTD features that are used when Oracle RTD is integrated with external applications.

For more detailed information about Oracle RTD, see [Related Documents](#) in the [Preface](#) chapter.

This chapter consists of the following topics:

- [Section 2.1, "Introduction to the Oracle RTD Decision Process"](#)
- [Section 2.2, "Integration Points Between External Applications and Oracle RTD"](#)
- [Section 2.3, "Overview of Oracle RTD Integration with External Applications"](#)
- [Section 2.4, "The Oracle RTD Decision Process"](#)
- [Section 2.5, "More About the Oracle RTD Decision Process Elements"](#)
- [Section 2.6, "General Oracle RTD Elements and Features"](#)
- [Section 2.7, "Oracle RTD Inline Services"](#)
- [Section 2.8, "Introduction to Oracle RTD Decision Studio"](#)
- [Section 2.9, "Analytic Reports and the Oracle RTD Decision Center"](#)

2.1 Introduction to the Oracle RTD Decision Process

The heart of Oracle RTD is a "decision engine" that helps users make decisions, by recommending the best options when they make their choices.

To illustrate the principles of the decision process and how these are incorporated in the Oracle RTD "decision engine", consider a common real-world decision: whether or not to accept a job offer from one of several companies that you have been investigating?

The data involved in the decision making process can be extensive. For example, a small subsection of job-related data to collect and evaluate could be:

- Company offering the job
- Job title

- Location
- Salary
- Promotion prospects

Company	Job	Location	Salary	Promotion Prospects
VeriLeaf	Quality Manager	Green Acres	220,000	Good
PlentiSol	Research Director	Balmington	250,000	Fair
FaunaFlex	Project Manager	North Elk	200,000	High

As well as gathering as much specific information as possible about the job, there are a number of key general questions that you as a prospective job hunter should address:

1. What are your choices?

As a simplification, assume that the choices in this example are to accept a single job offer, from one of the companies.

2. What are your goals?

You may have one or more goals that need to be compared and evaluated, for example:

- Minimize your daily travel time
- Maximize your financial compensation
- Improve your quality of life

3. What are the criteria for evaluating your choices against your goals?

In the real-life job-hunting situation, you typically have your own personal evaluation criteria, based on your requirements and past experiences. The process of evaluating your choices is often intuitive. However, the evaluation process can include satisfying more formal, numeric conditions, such as the requirement for a particular minimum salary.

In the Oracle RTD decision process, evaluation criteria are implemented by an ordering algorithm that prioritizes choices by assigning scores to them.

The scores for each Oracle RTD choice are computed using one or more of the following scoring methods:

- The rule driven scoring method uses explicit business rules, such as "Salary must be at least 200000" or "Promotion Prospect must be Good or High"
- The model driven scoring method uses implicit rules discovered from analyzing historical data stored in an Oracle RTD predictive model

When there are several performance goals for a decision, you can weight the goals. For example:

Performance Goal	Weighting
Maximize your daily travel time	30
Maximize your financial compensation	50
Improve your quality of life	20

Oracle RTD can score each choice against each performance goal or weighted combination of goals.

The net effect is that Oracle RTD provides a numeric score for each choice, such as in the following table:

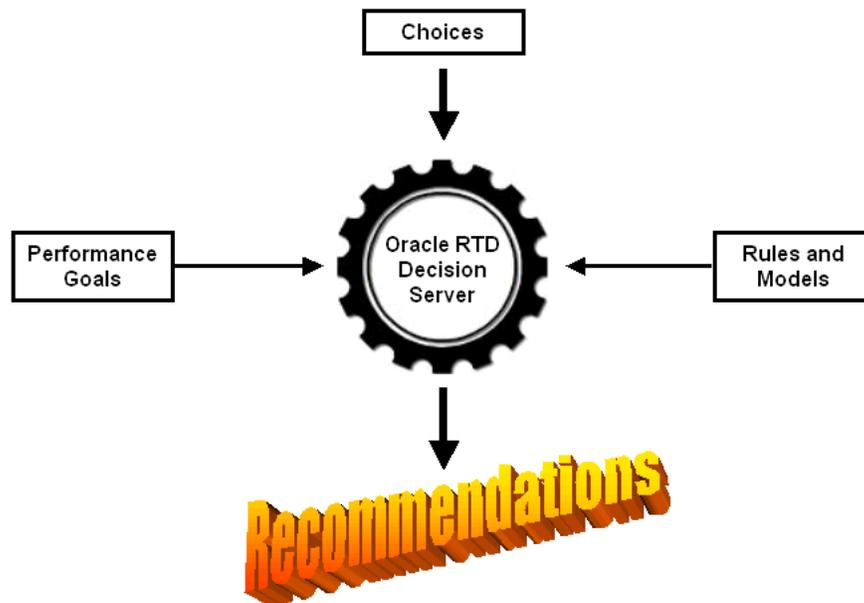
Company	Job	Location	Salary	Promotion Prospects	Choice	Score
VeriLeaf	Quality Manager	Green Acres	220,000	Good	Accept VeriLeaf job	60
PlentiSol	Research Director	Balmington	250,000	Fair	Accept PlentiSol job	50
FaunaFlex	Project Manager	North Elk	200,000	High	Accept FaunaFlex job	80

Oracle RTD Decision Making Features Overview

The overall principles and underlying elements described in the job hunting example are incorporated as basic features of Oracle RTD, as shown in the following table:

Questions in the Decision Making Process	Oracle RTD Features
What are your choices?	Choices
What are your goals?	Performance Goals
What are your criteria for evaluating those goals?	Rules and Models

The following diagram shows a high-level overview of how these features interact to fulfill the basic objective of Oracle RTD, namely to provide recommendations from a number of alternatives or choices.



For more information on these features and how to use them in Oracle RTD, see the following sections:

- [Section 2.4, "The Oracle RTD Decision Process"](#)
- [Section 2.5, "More About the Oracle RTD Decision Process Elements"](#)

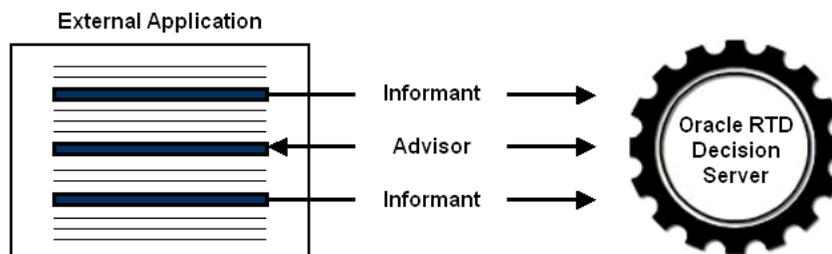
In general, Oracle RTD connects with other applications, and passes its recommendations to these external applications. See the next section for more information about how Oracle RTD integrates with external applications.

2.2 Integration Points Between External Applications and Oracle RTD

Applications that you develop to interact with Oracle RTD are referred to as external applications. Typically external applications consist of many processing steps and stages. The points at which external applications communicate with Oracle RTD are generically known as **Integration Points**.

There are two main types of Integration Point:

- An **Informant** is a process that passes data from the external application to Oracle RTD
- An **Advisor** is a two-way process, that both passes data from the external application to Oracle RTD, and also receives data sent back from Oracle RTD to the external application



Advisors are the main method by which an external application requests and receives recommendations from Oracle RTD.

Each external application can have many Informants and Advisors.

2.3 Overview of Oracle RTD Integration with External Applications

Many applications are based on a dialog with a user, which leads to the application presenting alternative strategies or choices to the user.

Typically, the dialog between the application and the user proceeds as follows:

1. The user starts a transaction.
2. The application retrieves information about the user.
3. Optionally, the user provides extra information concerning the transaction.
4. The application presents the user with one or more choices.
5. The user accepts or ignores the choice or choices.
6. Optionally, the previous two stages may be performed several times during the course of the transaction.
7. The user ends the transaction.

To determine which choices to present to a user, external applications can use various factors, such as:

- Profile information about the user
- Current information about the transaction
- The user’s preferences, if known
- Past activities or transactions associated with the user
- User access method, such as the Web or a custom interface
- Time of day, month, or season

Oracle RTD provides a set of tools that can analyze all these factors, and recommend the best choices to the external applications. Through these recommendations, Oracle RTD enables the companies that run the external applications to make better business decisions.

Figure 2–1 shows, in outline form, how a typical application interfaces with Oracle RTD.

Figure 2–1 Overview of External Application Integration with Oracle RTD

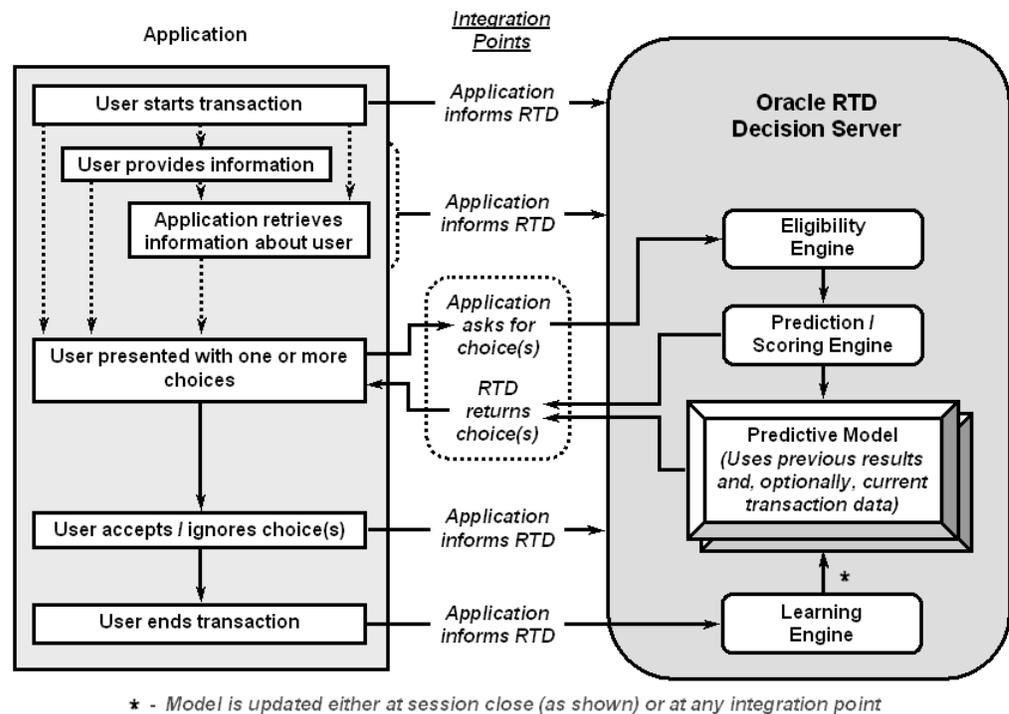


Figure 2–1 shows one Advisor and four Informants, the Informants corresponding to the following key stages in the application:

- The user has started the transaction.
- The external application has acquired more information about the user.
- The user has accepted or rejected a choice.
- The user has ended the transaction.

2.4 The Oracle RTD Decision Process

This section shows the general Oracle RTD decision process flow. For details about the Oracle RTD features and elements used in the process flow, see [Section 2.5, "More About the Oracle RTD Decision Process Elements."](#)

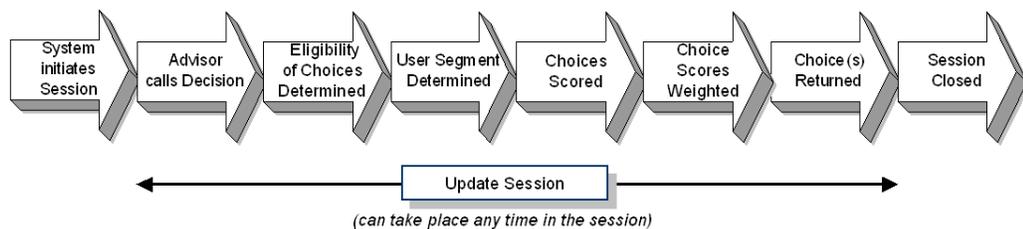
The Oracle RTD decision process is based on a framework that takes into account the following factors:

- The overall performance goals with which an organization is concerned
- The action required to score each of the available choices
- A weighting of the performance goals based on segments of the population

Decisions are called by Advisors to score Choices, and return one or more Choices from a Choice Group. The set up of a Decision must include at least one Choice Group from which Choices are selected, and a function or rule to score the Choices. At run time, the Decision collects all the eligible Choices that exist in each of the Choice Groups. Then, the Choices are scored to finally determine the ranked order to send back through the Advisor.

[Figure 2–2](#) shows the basic Oracle RTD processes, which include session start and finish, as well as the Oracle RTD decision process steps.

Figure 2–2 Oracle RTD Decision Process Flow



The steps represent the different stages in the overall process of acquiring the necessary data and processing a decision, as follows:

1. The system initiates the session.

When a user log on, and the external application connects to Oracle RTD, Oracle RTD establishes a session.

The external application generally acquires as much information about the user as possible, and passes it to Oracle RTD using one or more Informants.

Oracle RTD may also retrieve further information from Data Sources and Entities defined in the Inline Service associated with the external application.
2. The Advisor calls a Decision.

A request through an Advisor call initiates the decision process. The set of choices to evaluate for the decision is then determined for each of the associated Choice groups.
3. The eligibility of Choices is determined.

The eligibility rules for the Choices are invoked, to determine which Choices pass on to the next stage of the decision process.
4. The user segment is determined.

Filtering rules, if created, are then used to segment the user population. Based on the segment, the designated weightings for each of the Performance Goals is used in scoring each eligible Choice.

5. The Choices are scored.

All eligible Choices are scored for each associated Performance Goal.

6. The Choice scores are weighted.

Based on the segment, different weights are applied to the Performance Goal scores.

7. The Choice or Choices are returned to the external application.

Oracle RTD returns one or more Choices to the external application, passing Choice names and any designated Choice attribute that the external application needs. The requesting application then displays the Choices or processes the information accordingly.

8. The session information is updated.

This step can take place at any stage of the decision process. Its main effect is to update the Oracle RTD server with any new available information about the given session.

In addition, Models can be updated from the session information either at specified integration points or at the end of the session.

9. The session is closed.

The active Oracle RTD session is closed and any wrap up logic is executed, including learning on any Models defined to learn at session close.

2.5 More About the Oracle RTD Decision Process Elements

This section provides more details about the following Oracle RTD elements used in the decision processing framework:

- [Performance Goals](#)
- [Decisions](#)
- [Choice Groups and Choices](#)
- [Eligibility Rules](#)
- [Filtering Rules](#)
- [Scoring Rules](#)
- [Models](#)

2.5.1 Performance Goals

Designers creating a decision process for an organization must consider the overall Performance Goals of the organization. Performance Goals consist of the specific metrics with which the organization has chosen to measure its success. These goals are then associated to choice groups and decisions to identify how each choice will be scored against those them. Some common performance metrics are revenue, costs, number of products per customer, and customer satisfaction.

If you set more than one Performance Goal in an Inline Service, you must specify the relative importance of each one by assigning normalization factors for each Performance Goal.

2.5.2 Decisions

Decisions score eligible Choices and rank them based on the weightings given for associated Performance Goals.

Oracle RTD supports the following types of Decisions:

- Rule-driven Decisions
- Model-driven Decisions
- Hybrid Decisions

Rule-driven Decisions are defined in business related terms expressed by business users. An example could be the business rule not to sell credit cards to customers when their credit rating is low.

Model-driven Decisions are driven by scores calculated and determined by Models formed from empirical data. An example could be the decision to present an Overdraft Protection offer to a call center user who lives in California, whose occupation is graphical artist, and who has called to change his address. Based on its previous learnings, the model has determined that similar users are 61% likely to accept the Overdraft Protection offer.

Hybrid Decisions use the scoring methods of both the Rule-driven and the Model-driven decisions.

In general, each Decision may be associated with:

- One or more Choice Groups
- One or more Performance Goals
- One or more segments of the user population, where each segment can have different weightings for each Performance Goal

2.5.3 Choice Groups and Choices

Choice Groups and Choices are the Inline Service elements from which Decisions draw their possible Choices, and which become targets of analysis for Choice and Choice Event Models.

Choice Groups and Choices form a hierarchy, where:

- Each Choice belongs to one Choice Group only
- Each Choice Group can have one or more sub-Choice Groups

Choices exist only at the lowest level of a Choice Group hierarchy branch.

Choices can be used by a Decision, so that they can be returned by Advisors, and can be registered to either Choice or Choice Event models through Informants.

Choice Group and Choice Attributes

Choice Groups and Choices have attributes, that is, data used in the processing and presentation of Choices.

Typically, you define the attributes of Choices at a higher Choice Group level, where you can also specify default values for the attributes. The Choice Group attributes are

inherited by lower level Choice Groups and Choices. You can override default values at the lower levels.

Static and Dynamic Choices

Choices can either be static or dynamic.

Static Choices are explicitly defined in the Inline Service.

Dynamic Choices are Choices that are stored and maintained in an external application, such as promotions stored in a separate marketing application. When required for the decision process, Dynamic Choices are retrieved from the external application. The mechanisms for retrieving and using Dynamic Choices are defined in the Inline Service, but the actual Dynamic Choice values may vary for each user session.

For more information about Dynamic Choices, see *Oracle Real-Time Decisions New Features Guide Version 2.2.1*.

2.5.4 Eligibility Rules

Choices and choice groups have rules that determine their eligibility to participate in a decision.

You can define eligibility rules at the Choice Group and Choice levels.

Choices inherit rules from higher levels, and may also have their own rules. At each level, a logical AND is performed between the higher-level rules and the current-level rule, with the result placed in the current-level element.

2.5.5 Filtering Rules

Choices and Choice Groups can use filtering rules as another form of eligibility. In addition, a filtering rule can also be used to segment the user population for which Decisions are being made, and controls the effect of each Performance Goal associated to the Decision.

2.5.6 Scoring Rules

Scoring rules are similar in setup to eligibility rules, but rather than evaluate the rule to a TRUE/FALSE outcome, a numeric score is returned instead. A score can be computed for a given Performance Goal tied to a Choice, and can affect the rank of the Choices in the decision process.

2.5.7 Models

There are two standard types of model in Oracle RTD:

- Choice Model
- Choice Event Model

Each Choice Model or Choice Event Model is always associated with a single Choice Group.

Both types of model can be used for prediction and for generating analysis reports.

2.5.7.1 Model Usage Summary

Input

The main objective of any model is to show, for each choice of the associated choice group, what factors influenced a particular choice.

Models are updated with, and "learn" from the following data:

- All the data in the Session entity - specifically, all the Session attributes, unless specifically excluded from analysis through the Inline Service configuration

In addition, Choice Event Models also require event-related details. For more information, see [Section 2.5.7.3, "Choice Event Models."](#)

The update and learning process happens in a transaction either at session close or at any integration point.

Outputs

Both types of model can be used for prediction and for generating analysis reports.

The outputs generated directly and indirectly from a model are as follows:

- Model scores for a given Choice which can be used as part of the decision process
- Analytic reports in the Decision Center
- Model snapshots that enable users to generate their own analytic reports

2.5.7.2 Choice Models

The main objective of a Choice Model is, for each choice, to derive meaningful information from the data associated simply with the choice itself. A Choice Model does not need the extra dimension of base and positive outcome events, which are required for Choice Event Models.

For instance, in a call center application, one of the key data elements is the reason for a call. After collecting more information about the call and the caller, you can provide this information to a Call Reason Choice Model, and then use this in Oracle RTD Decision Center to analyze and compare the driving attributes of different call reasons.

Another example of a Choice Model is an Abandonment Model, with two choices, Abandoned and Not Abandoned. For both choices, the model stores data associated with the user and the transaction, and whether the user abandoned the transaction before completion. You can use the model not only to analyze potential abandonment factors, but also to predict the likelihood of whether subsequent users will abandon their transactions.

2.5.7.3 Choice Event Models

For each Oracle RTD Choice Event Model, in addition to specifying a Choice Group, you must also specify one Base event and one or more Positive Outcome events.

In the simplest case, there are two significant events in a transaction, the presentation of a choice and the acceptance of the choice.

In Oracle RTD, events are defined at the Choice Group level, and selected within the Model to describe "base" and "success" parameters.

For each Choice Event Model, you must define:

- One Base event, used as the base for analysis.

Typically, this is the event associated with the presentation of the choice.

- One or more Positive Outcome events, each of which indicate a successful prediction.

Typically, the standard positive outcome event is the event associated with the acceptance of the choice.

2.6 General Oracle RTD Elements and Features

Some Oracle RTD objects have a general usage within and across Oracle RTD processes. This section describes the following general Oracle RTD elements and features:

- [Data Sources](#)
- [Entities](#)
- [Functions](#)

2.6.1 Data Sources

A Data Source is configured in an Inline Service to access data from an outside source. The structure and format of Data Sources can vary, as follows:

- The rows and columns of a database table or view
- The output values and result row sets from a stored procedure

A Data Source can be configured to retrieve either a single record or multiple rows.

Each Data Source contains Input and Output columns:

- The Input columns are used in the WHERE clause of the query to the Data Source to select the rows to retrieve.
- The Output columns are the data that is retrieved and used by Oracle RTD in the decision process.

2.6.2 Entities

An Entity is a logical representation of data, that can be populated from one or more Data Sources, through data retrieved by an Integration Point, or through functional derivations. Entities are the data objects that can be used by the other Oracle RTD elements, and form a logical level of abstraction from Data Sources and Integration Points.

An Entity is a set of named attributes and methods to access the attributes. One attribute per Entity is usually designated as the Entity key.

An attribute of an Entity is analogous to a column of a database table, with one important distinction: an Oracle RTD attribute may consist either of one value or many values. The type of attribute that can have multiple values is called an Array attribute.

The integration of Entities and their component attributes to the appropriate data is implemented by mapping. You can explicitly map Entity attributes to Data Source columns, or you can implicitly map them through the use of Java functions that populate the Entity attributes.

An Entity, while it contains its own attributes, may also be an attribute of another Entity. For example, a customer can have many orders. In Oracle RTD, you can define Customers and Orders as separate Entities, mapped from corresponding Data Sources. You can then specify the Orders Entity to be an attribute of the Customers Entity.

Session Entities

The Session is the fundamental Oracle RTD unit of runtime data. Data is kept in memory for the duration of the Session. Every Inline Service contains one Session Entity.

For a Model to be able to learn from the attributes of a non-Session Entity, that Entity must be defined as an attribute of the Session Entity.

For example, in an Inline Service, you can define Customer, Call, and Product as logical Entities, and then add these as attributes to the Session Entity, so that the Oracle RTD server can use these Entities as inputs to the Models.

2.6.3 Functions

Functions, written in Java, provide extra processing capabilities to many Oracle RTD elements. For example, selection functions can be used by decisions as a custom way to make a choice.

Functions can also serve as general-purpose code, for example, to determine date differences, or to convert data into different data types.

Other users of functions include:

- Populating derived entity attributes
- Comparing values in choice eligibility rules
- Retrieving lookup values
- Writing to log files

Functions may also call other functions.

2.7 Oracle RTD Inline Services

An Oracle RTD Inline Service consists of all the Oracle RTD elements necessary to interface with an external application and model the desired business process.

The main elements of an Inline Service are the following:

- Data Sources
- Entities
- Integration Points
- Choice Groups and Choices
- Decisions
- Filtering Rules
- Scoring Rules
- Models
- Performance Goals
- Functions

Not all Inline Services have all of these elements. The specific requirements of each external application determine which elements are needed in the associated Inline Service.

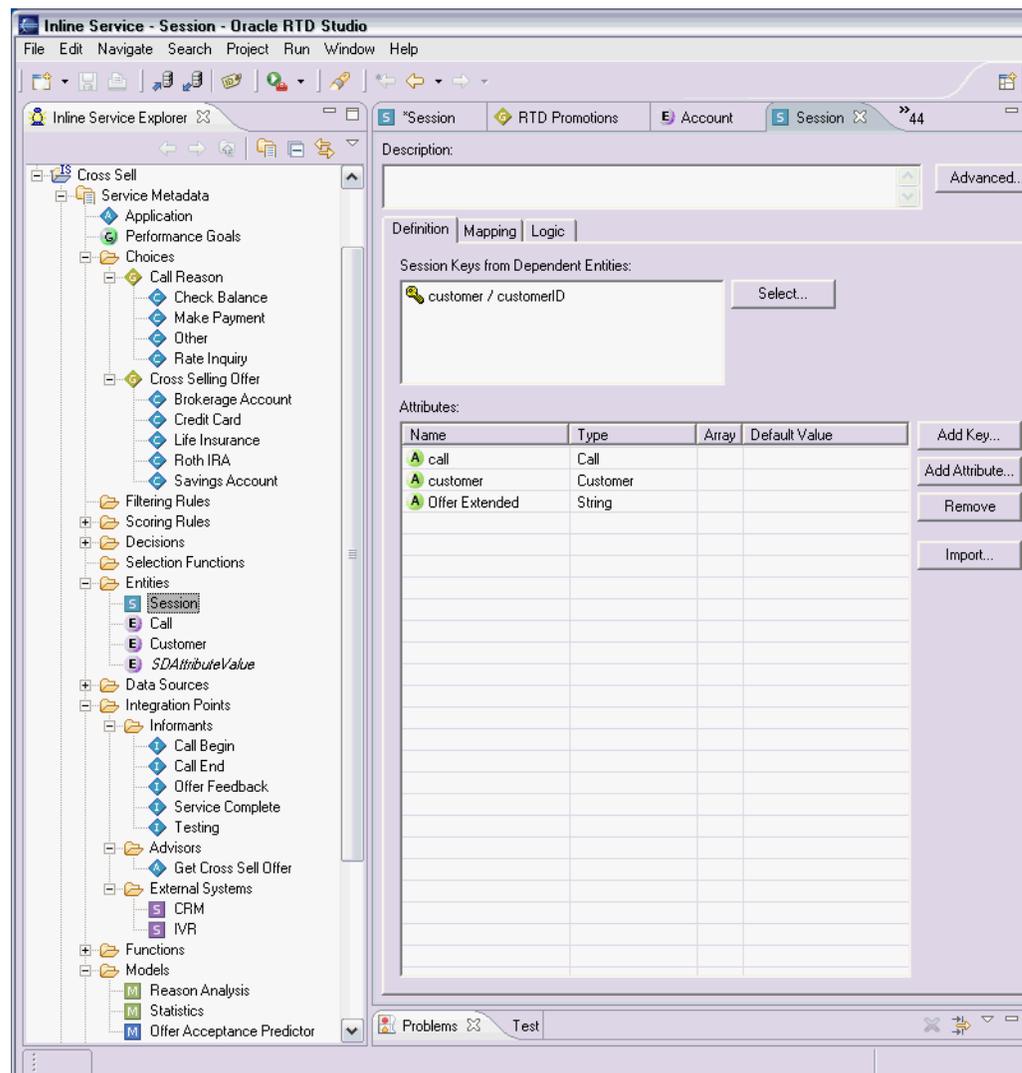
For more information on how to configure an Inline Service, see the [Related Documents](#) in the [Preface](#) chapter, in particular the *Oracle Real-Time Decisions Decision Studio Reference Guide*.

2.8 Introduction to Oracle RTD Decision Studio

In Oracle RTD, you define the Inline Service elements in the Oracle RTD Decision Studio. You must first configure, then deploy an Inline Service to the Oracle RTD server before the Inline Service can be used by an application.

[Figure 2–3](#) shows some of the elements of an Inline Service, called Cross Sell, as displayed in the Decision Studio.

Figure 2–3 Example of an Inline Service Displayed in Decision Studio



For more information about how to define and deploy Inline Services, see *Oracle Real-Time Decisions Decision Studio Reference Guide*.

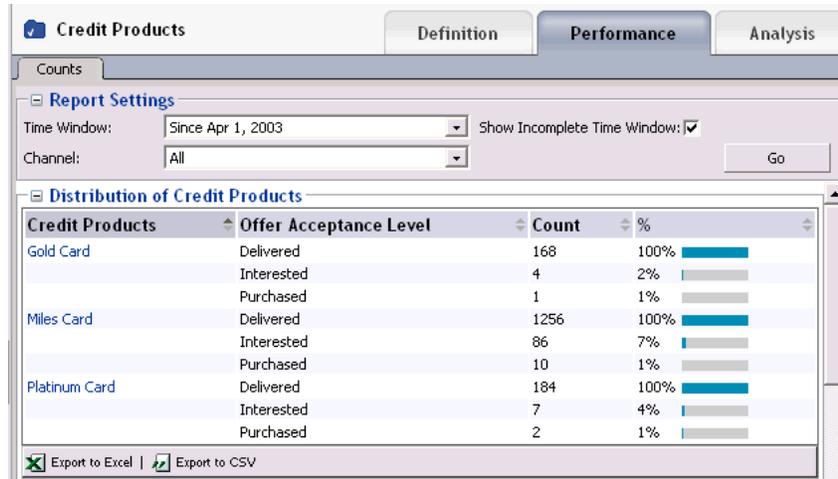
2.9 Analytic Reports and the Oracle RTD Decision Center

Oracle RTD Decision Center is a client tool for business users to explore, analyze, examine, and even modify the structure and data gathered by a deployed Inline Service.

The Oracle RTD Decision Center provides a variety of analytic reports, both for performance analysis and model analysis.

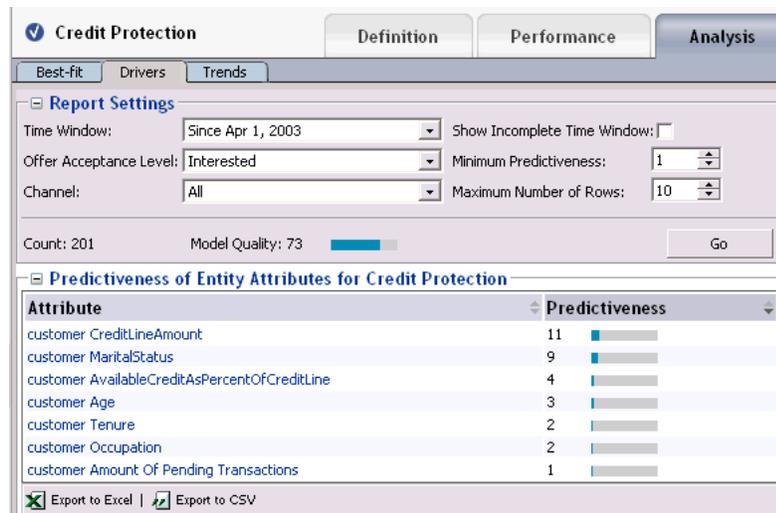
For example, there are several reports at the choice group and choice level, such as the following examples from a Cross Sell application:

- **Choice Group Performance Counts**



The Choice Group Performance Counts shows the total counts for each choice or choice event occurrence in a choice group.

- **Choice Analysis Drivers**



The Choice Analysis Drivers report identifies the attributes that are influential as drivers of predictiveness for each of the choices.

Predictiveness is a measure of the relationship strength between entity attributes, that are the model input, and choice and choice events, that are the model output.

A drilldown on any of the attribute hyperlinks will reveal additional reports about the attribute values themselves.

- **Choice Analysis Trends**

The screenshot shows the 'Credit Protection' report in the 'Analysis' tab. The 'Report Settings' section includes filters for time windows (Jul 1, 2003 - Sep 30, 2003 and Since Oct 1, 2003), offer acceptance level (Interested), and channel (All). Below the settings, a table displays 'Changes in Predictiveness of Entity Attributes for Credit Protection'.

Attribute	Predictiveness Jul 1, 2003 - Sep 30, 2003	Predictiveness Since Oct 1, 2003	% Change
customer Age	4	5	18%
customer AvailableCreditAsPercentOfCreditLine	7	8	7%
customer CreditLineAmount	12	13	6%
customer MaritalStatus	10	10	-7%

The Choice Analysis Trends report shows the change of predictiveness for each of the attributes for a choice over two selected model time windows.

- **Choice Analysis Best Fit**

The screenshot shows the 'Credit Protection' report in the 'Analysis' tab. The 'Report Settings' section includes filters for time window (Since Apr 1, 2003), offer acceptance level (Interested), and channel (All). Below the settings, a table displays 'Highest correlating attribute values for Credit Protection'.

Attribute	Value	Correlation
customer CreditLineAmount	6000 to 20000	High
customer MaritalStatus	Divorced	High
customer AvailableCreditAsPercentOfCreditLine	14 to 26	High
customer Amount Of Pending Transactions	2401 to 9995	High
customer Tenure	0	High
customer CardType	Student	High
customer Age	43 to 47	High
customer CallsLast6Months	3 to 10	High
customer MinimumAmountDue	1 to 100	High
customer DayOfWeek	Tuesday	High

The Choice Analysis Best Fit report shows all the attributes and values that are most likely to predict the specified event outcome.

Oracle RTD also provides a variety reports that show the effectiveness of entities and entity attributes for predicting choices.

For more information about how to view, analyze, and modify the structure and data of Inline Services in the Decision Center, see *Oracle Real-Time Decisions Decision Center User Guide*.

Part II

Base E-Commerce Inline Service

Part 2 describes the component elements of the released Base E-Commerce Inline Service, and provides general guidelines for how to configure the Inline Service to fulfill an organization's business requirements.

Part 2 contains the following chapters:

- [Chapter 3, "Base E-Commerce Inline Service Elements"](#)
- [Chapter 4, "Configuring the Base E-Commerce Inline Service"](#)

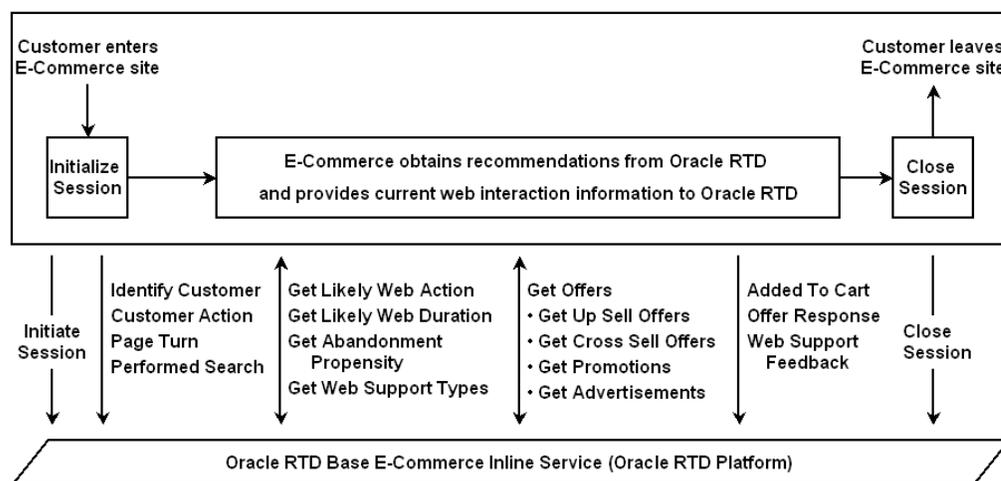
Base E-Commerce Inline Service Elements

This chapter describes the elements in the Base E-Commerce Inline Service. It contains the following topics:

- [Section 3.1, "Introduction to the Base E-Commerce Inline Service"](#)
- [Section 3.2, "Entity Logical Model"](#)
- [Section 3.3, "Choices Groups and Choices"](#)
- [Section 3.4, "Inline Service Advisors"](#)
- [Section 3.5, "Inline Service Informants"](#)
- [Section 3.6, "Inline Service Entities"](#)
- [Section 3.7, "Inline Service Functions"](#)

3.1 Introduction to the Base E-Commerce Inline Service

The following diagram shows an overview of the stages of a typical E-Commerce application and its possible interactions with the Base E-Commerce Inline Service.



The Base E-Commerce Inline Service serves as a general E-Commerce framework for customers to adapt to their business processes.

The Base E-Commerce Inline Service provides pre-defined entities, choices, decisions, models, and integration points.

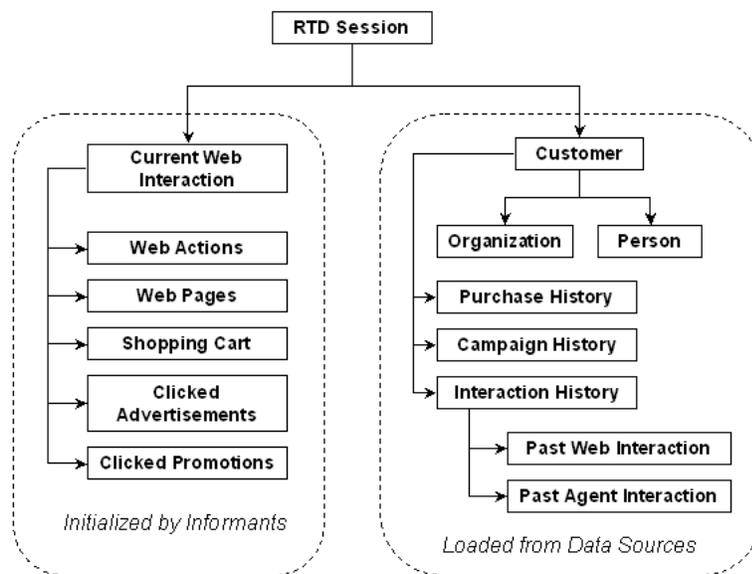
From an end user perspective, the Base E-Commerce Inline Service is designed on the assumption that customers will perform additional configuration and further customization to fulfill their business requirements.

Additional tasks would involve mapping the logical entity attributes to a customer's physical data sources and to develop the front-end environment for the presentation of any Oracle RTD recommendations.

For more information, see [Chapter 4, "Configuring the Base E-Commerce Inline Service."](#)

3.2 Entity Logical Model

The following diagram illustrates the overall general logical entity object model of the Base E-Commerce Inline Service.



For more information about the included Entities, see [Section 3.6, "Inline Service Entities."](#)

3.3 Choices Groups and Choices

This section presents an overview of the choice groups and choices, and their usage in the integration points.

This section contains the following topics:

- [Section 3.3.1, "Offer-Oriented Choices"](#)
- [Section 3.3.2, "Interaction-Oriented Choices"](#)
- [Section 3.3.3, "Informants Usage with Offer-Oriented and Transaction-Oriented Choices"](#)

For more information about how the choices and choice groups are used, see [Section 3.4, "Inline Service Advisors."](#)

3.3.1 Offer-Oriented Choices

Choices for offer-oriented use cases are structured as follows:



Offers and all other choice groups under it are comprised of dynamic choices. The choice data is either supplied as an Advisor's incoming parameter value (see [Section 3.4, "Inline Service Advisors"](#)) or retrieved by Oracle RTD from external data sources.

Advisors for Offer-Oriented Choices

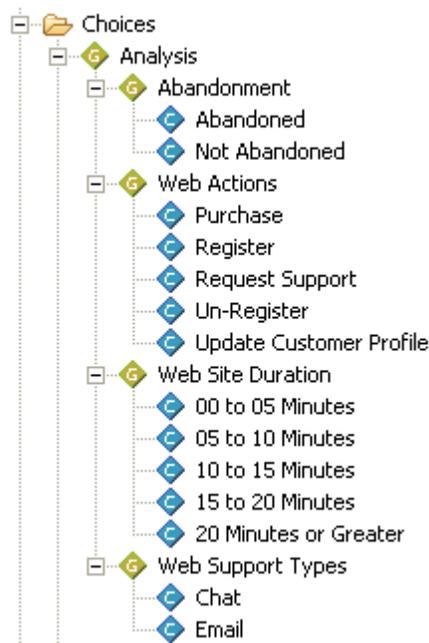
The offer-oriented choices are returned when invoking the following Advisors:

- **Get Cross Sell Offers**
- **Get Upsell Offers**
- **Get Promotions**
- **Get Advertisements**
- **Get Offers**
 - Updates current web interaction attributes and Web Pages for the session.
 - Identifies any given page a customer navigates to that needs to be tracked.

For more specific information about the Advisors, see [Section 3.4, "Inline Service Advisors."](#)

3.3.2 Interaction-Oriented Choices

Choice groups and choices for interaction-oriented usages are structured as follows. Note that the choices under each choice group can be reconfigured to suit the end users needs and serve as a template.



- **Abandonment**

- Identifies if customers have abandoned their web session. This would be determined by the business based on their definitions of abandonment.

Abandonment choice can be obtained by invoking the Advisor **Get Abandonment Propensity**, and its prediction model is updated by invoking the Advisor **Close Session**.

- Returns likelihood of abandonment for a customer.

- **Web Actions**

- Identifies key actions done by a customer that can later be applied to entity attributes or models, for example, Update Customer Profile, Register, Un-register. Chat request.

Web Actions choice can be obtained by invoking the Advisor **Get Likely Web Action**, and its prediction model is updated by invoking the Informant **Customer Action**.

- **Web Site Duration**

- Web Site Durations’s prediction model is updated by invoking the Informant **Close Session**.

- **Web Support Types**

- Web Support Types choice can be obtained by invoking the Advisor **Get Web Support Types**, and its prediction model is updated by invoking the Informant **Web Support Feedback**.
- Returns likelihood for a customer to accept chat request or email support if offered.

3.3.3 Informants Usage with Offer-Oriented and Transaction-Oriented Choices

This section describes the following usages of the Base E-Commerce Inline Service Informants:

- How the Informants provide feedback and additional information to Oracle RTD for the offer-oriented use cases described in [Section 3.3.1, "Offer-Oriented Choices"](#)
- How the Informants are used for updating the choice groups described in [Section 3.3.2, "Interaction-Oriented Choices"](#)

For more specific information about the Informants, see [Section 3.5, "Inline Service Informants."](#)

- **Initiate Session**
 - Updates current web interaction attributes for the session.
- **Identify Customer**
 - Updates customer related attributes for the session.
 - Provides Oracle RTD with customer id and profile information if a customer is identified while on the web site. In most cases, identification can be made upon user logon or by previously set cookies at the customer's browser.
- **Page Turn**
 - Updates current web interaction attributes and Web Pages for the session.
 - Identifies any given page a customer navigates to that needs to be tracked.
- **Performed Search**
 - Updates current web interaction attributes and Search Keywords for the session.
 - Identifies search words used by a customer.
- **Customer Action**
 - Updates current web interaction attributes and Web Action for the session.
 - Identifies key actions performed by a customer that can later be applied to entity attributes or models, for example, Update Customer Profile, Register, Un-register, Chat request.
- **Added To Cart**
 - Updates current web interaction attributes and Shopping Cart Items for the session.
- **Offer Response**
 - Creates prediction model entries and also captures shopping cart addition events.
 - Identifies if an offer (Upsell, Cross Sell, Promotion, or Ad) is clicked, added to a cart, or purchased by a web user or customer.
- **Close Session**
 - Creates analytical prediction model entries and triggers learning.
 - Formally closes out the Oracle RTD session.

3.4 Inline Service Advisors

For each Advisor listed in this section, a detailed breakdown is provided for the Integration Point, followed by:

- The Decision called by the Advisor

- The Choice Group used by the Decision
- As appropriate, the Model associated with the Choice Group

3.4.1 Decisioning Advisors

This section describes the following advisors:

- [Get Upsell Offers](#)
- [Get Cross Sell Offers](#)
- [Get Promotions](#)
- [Get Advertisements](#)
- [Get Offers](#)

3.4.1.1 Get Upsell Offers

The Advisor Get Upsell Offers determines the likelihood for a customer to accept an upsell offer.

[Table 3–1](#) describes the parameters for the Advisor Get Upsell Offers.

Table 3–1 Advisor Get Upsell Offers

Parameter	Description
Advisor Name	Get Upsell Offers
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Product Id (optional) Number of Offers Rank Offers (array, optional, to be ranked, as source for dynamic choice)
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Upsell Offers
Group Decision	Select Upsell Offers
Default Choices	None
Logic	The supplied product id is assigned to Session / Supplied Product Id. This will be used by function: Select Product Id (for Cross Sell / Up Sell). If supplied, override the default number of choices returned by decision using the Number of Offers incoming parameter. If Rank Offer (string array) is supplied, set these into session as dynamic choice objects to be used by decision later.
Pre-condition	Base Product Id for Cross sell / Up sell must exist. Function: Select Product Id tries to find this Product Id from the following variables: 1 - Session / Supplied Product Id 2 - Session / Current Web Interaction / Current Viewed Product Id 3 - Session / Current Web Interaction / Last Added (into shopping cart) Product Id

Table 3–2 describes the parameters for the decision for the Advisor Get Upsell Offers.

Table 3–2 Decision for Advisor Get Upsell Offers

Parameter	Description
Decision Name	Select Upsell Offers
Select Choices From	Upsell Offers
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 33% Maximize Revenue 33% Popularity 33%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

Table 3–3 describes the parameters for the choice group for the Advisor Get Upsell Offers.

Table 3–3 Choice Group for Advisor Get Upsell Offers

Parameter	Description
Choice Group Name	Upsell Offers (Choice Group)
Choice Attributes	Product Id Product Name Product Line Product Type Product Category Offer Type = "Up Sell" Likelihood
Scores	Maximize Acceptance Likelihood = Predicted by Upsell Purchase Model: Purchased Maximize Revenue = Dynamic Choice / Unit Price Popularity = Dynamic Choice / Popularity Rank
Choice Events	Presented (inherited from Offers) Interested (inherited from Offers) Added To Cart Purchased
Choice Eligibility	None
Group Attributes	Upsell Products – Type=Product (Array) – Loading: Get Up Sell Product List (Select Product Id())
Group Eligibility	None
Dynamic Choices	Choice Id is Product Id

Table 3–4 describes the parameters for the model for the Advisor Get Upsell Offers.

Table 3–4 Model for the Advisor Get Upsell Offers

Parameter	Description
Model Name	Upsell Purchase Model (Choice Event Model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Upsell Offers
Base Event	Presented
Positive Events	Purchased
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	Offer Response Get Upsell Offers Get Offers
Temporary Data Storage	None

Dependencies:

- Informant - Initiate Session
- Informant - Identify Customer
- Informant - Page Turn
- Informant - Performed Search
- Informant - Customer Action
- Informant - Offer Response
- Informant - Added to Cart
- Informant - Close Session
- Choice Group - Offers
- Function - Select Product Id ()
- Function - Get Up Sell Product List (Product Id)

3.4.1.2 Get Cross Sell Offers

The Advisor Get Cross Sell Offers determines the likelihood for a customer to accept a cross sell offer.

Table 3–5 describes the parameters for the Advisor Get Cross Sell Offers.

Table 3–5 Advisor Get Cross Sell Offers

Parameter	Description
Advisor Name	Get Cross Sell Offers
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Product Id (optional) Number of Offers Rank Offers (array, optional, to be ranked, as source for dynamic choice)

Table 3–5 (Cont.) Advisor Get Cross Sell Offers

Parameter	Description
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Cross Sell Offers
Group Decision	Select Cross Sell Offers
Default Choices	None
Logic	The supplied product id is assigned to Session / Supplied Product Id. This will be used by function: Select Product Id (for Cross Sell / Up Sell). If supplied, override the default number of choices returned by decision using the Number of Offers incoming parameter. If Rank Offer (string array) is supplied, set these into session as dynamic choice objects to be used by decision later.
Pre-condition	Base Product Id for Cross sell / Up sell must exist. Function: Select Product Id tries to find this Product Id from the following variables: 1 - Session / Supplied Product Id 2 - Session / Current Web Interaction / Current Viewed Product Id 3 - Session / Current Web Interaction / Last Added (into shopping cart) Product Id

Table 3–6 describes the parameters for the decision for the Advisor Get Cross Sell Offers.

Table 3–6 Decision for Advisor Get Cross Sell Offers

Parameter	Description
Decision Name	Select Cross Sell Offers
Select Choices From	Cross Sell Offers
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 33% Maximize Revenue 33% Popularity 33%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

Table 3–7 describes the parameters for the choice group for the Advisor Get Cross Sell Offers.

Table 3–7 Choice Group for Advisor Get Cross Sell Offers

Parameter	Description
Choice Group Name	Cross Sell Offers (Choice Group)
Choice Attributes	Product Id Product Name Product Line Product Type Product Category Offer Type = "Cross Sell" Likelihood
Scores	Maximize Acceptance Likelihood = Predicted by Cross Sell Purchase Model: Purchased Maximize Revenue = Dynamic Choice / Unit Price Popularity = Dynamic Choice / Popularity Rank
Choice Events	Presented (inherited from Offers) Interested (inherited from Offers) Added To Cart Purchased
Choice Eligibility	None
Group Attributes	Cross Sell Products - Type=Product (Array) - Loading: Get Cross Sell Product List (Select Product Id())
Group Eligibility	None
Dynamic Choices	Choice Id is Product Id

Table 3–8 describes the parameters for the model for the Advisor Get Cross Sell Offers.

Table 3–8 Model for Advisor Get Cross Sell Offers

Parameter	Description
Model Name	Cross Sell Purchase Model (Choice Event Model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Cross Sell Offers
Base Event	Presented
Positive Events	Purchased
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	Offer Response Get Cross Sell Offers Get Offers
Temporary Data Storage	None

Dependencies:

- Informant - Initiate Session

- Informant - Identify Customer
- Informant - Page Turn
- Informant - Performed Search
- Informant - Customer Action
- Informant - Offer Response
- Informant - Added to Cart
- Informant - Close Session
- Choice Group - Offers
- Function - Select Product Id ()
- Function - Get Cross Sell Product List (Product Id)

3.4.1.3 Get Promotions

The Advisor Get Promotions determines the likelihood for a customer to have an interest on a presented promotion.

[Table 3–9](#) describes the parameters for the Advisor Get Promotions.

Table 3–9 Advisor Get Promotions

Parameter	Description
Advisor Name	Get Promotions
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Number of Offers Rank Offers (array, optional, to be ranked, as source for dynamic choice)
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Promotions
Group Decision	Select Promotions
Default Choices	None
Logic	If supplied, override the default number of choices returned by decision using the Number of Offers incoming parameter. If Rank Offer (string array) is supplied, set these into session as dynamic choice objects to be used by decision later.
Pre-condition	None

[Table 3–10](#) describes the parameters for the decision for the Advisor Get Promotions.

Table 3–10 Decision for Advisor Promotions

Parameter	Description
Decision Name	Select Promotions
Select Choices From	Promotions
Number of Choices to Select	5

Table 3–10 (Cont.) Decision for Advisor Promotions

Parameter	Description
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

Table 3–11 describes the parameters for the choice group for the Advisor Get Promotions.

Table 3–11 Choice Group for Advisor Promotions

Parameter	Description
Choice Group Name	Promotions (Choice Group)
Choice Attributes	Promotion Id Promotion Name Promotion Type Promotion Period Offer Type = "Promotion" Likelihood
Scores	Maximize Acceptance Likelihood = Predicted by Promotion Interest Model: Interested
Choice Events	Presented (inherited from Offers) Interested (inherited from Offers)
Choice Eligibility	None
Group Attributes	Promotion List - Type=Promotion (Array) - Data loading function: Get Promotion List ()
Group Eligibility	None
Dynamic Choices	Choice Id is Promotion Id

Table 3–12 describes the parameters for the model for the Advisor Get Promotions.

Table 3–12 Model for Advisor Get Promotions

Parameter	Description
Model Name	Promotion Interest Model (Choice Event Model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Promotions
Base Event	Presented
Positive Events	Interested
Partitioning Attributes	None
Excluded Attributes	None

Table 3–12 (Cont.) Model for Advisor Get Promotions

Parameter	Description
Learn Location	Offer Response Get Promotions Get Offers
Temporary Data Storage	None

Dependencies:

- Informant - Initiate Session
- Informant - Identify Customer
- Informant - Page Turn
- Informant - Performed Search
- Informant - Customer Action
- Informant - Offer Response
- Informant - Added to Cart
- Informant - Close Session
- Choice Group - Offers

3.4.1.4 Get Advertisements

The Advisor Get Advertisements determines the likelihood for a customer to have an interest on a presented advertisement.

Table 3–13 describes the parameters for the Advisor Get Advertisements.

Table 3–13 Advisor Get Advertisements

Parameter	Description
Advisor Name	Get Advertisements
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Number of Offers Rank Offers (array, optional, to be ranked, as source for dynamic choice)
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Advertisements
Group Decision	Select Advertisements
Default Choices	None
Logic	If supplied, override the default number of choices returned by decision using the Number of Offers incoming parameter. If Rank Offer (string array) is supplied, set these into session as dynamic choice objects to be used by decision later.
Pre-condition	None

Table 3–14 describes the parameters for the decision for the Advisor Get Advertisements.

Table 3–14 Decision for Advisor Get Advertisements

Parameter	Description
Decision Name	Select Advertisements
Select Choices From	Advertisements
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("Presented");}

Table 3–15 describes the parameters for the choice group for the Advisor Get Advertisements.

Table 3–15 Choice Group for Advisor Get Advertisements

Parameter	Description
Choice Group Name	Advertisements (Choice Group)
Choice Attributes	Ad Id Ad Name Ad Type Ad Category Offer Type = "Ad" Likelihood
Scores	Maximize Acceptance Likelihood = Predicted by Advertisement Interest Model: Interested
Choice Events	Presented (inherited from Offers) Interested (inherited from Offers)
Choice Eligibility	None
Group Attributes	Ad List - Type=Ad (Array) - Data loading function: Get Ad List ()
Group Eligibility	None
Dynamic Choices	Choice Id is Ad Id

Table 3–16 describes the parameters for the model for the Advisor Get Advertisements.

Table 3–16 Model for Advisor Get Advertisements

Parameter	Description
Model Name	Advertisement Interest Model (Choice Event Model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian

Table 3–16 (Cont.) Model for Advisor Get Advertisements

Parameter	Description
Choice Group	Promotions
Base Event	Presented
Positive Events	Interested
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	Offer Response Get Advertisements Get Offers
Temporary Data Storage	None

Dependencies:

- Informant - Initiate Session
- Informant - Identify Customer
- Informant - Page Turn
- Informant - Performed Search
- Informant - Customer Action
- Informant - Offer Response
- Informant - Added to Cart
- Informant - Close Session
- Choice Group - Offers

3.4.1.5 Get Offers

The Advisor Get Offers determines the likelihood for a customer to accept an offer.

The Advisor Get Offers returns a mix of offers:

- Upsell Offers
- Cross Sell Offers
- Promotions
- Advertisements

[Table 3–17](#) describes the parameters for the Advisor Get Offers.

Table 3–17 Advisor Get Offers

Parameter	Description
Advisor Name	Get Offers
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Product Id (optional) Number of Offers Rank Offers (array, optional, to be ranked, as source for dynamic choice)

Table 3–17 (Cont.) Advisor Get Offers

Parameter	Description
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Offers
Group Decision	Select Offers
Default Choices	None
Logic	The supplied product id is assigned to Session / Supplied Product Id. This will be used by function: Select Product Id (for Cross Sell / Up Sell). If supplied, override the default number of choices returned by decision using the Number of Offers incoming parameter. If Rank Offer (string array) is supplied, set these into session as dynamic choice objects to be used by decision later.
Pre-condition	Base Product Id for Cross sell / Up sell must exist. Function: Select Product Id tries to find this Product Id from the following variables: 1 - Session / Supplied Product Id 2 - Session / Current Web Interaction / Current Viewed Product Id 3 - Session / Current Web Interaction / Last Added (into shopping cart) Product Id

Table 3–18 describes the parameters for the decision for the Advisor Get Offers.

Table 3–18 Decision for Advisor Get Offers

Parameter	Description
Decision Name	Select Offers
Select Choices From	Offers
Number of Choices to Select	5
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Acceptance Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("presented");}

Table 3–19 describes the parameters for the choice group for the Advisor Get Offers.

Table 3–19 Choice Group for Advisor Get Offers

Parameter	Description
Choice Group Name	Offers (Choice Group)
Choice Attributes	Offer Type Likelihood

Table 3–19 (Cont.) Choice Group for Advisor Get Offers

Parameter	Description
Scores	Maximize Acceptance Likelihood
Choice Events	Presented Interested
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Dependencies:

- Informant - Initiate Session
- Informant - Identify Customer
- Informant - Page Turn
- Informant - Performed Search
- Informant - Customer Action
- Informant - Offer Response
- Informant - Added to Cart
- Informant - Close Session
- Choice Group - Upsell Offers
- Choice Group - Cross Sell Offers
- Choice Group - Promotions
- Choice Group - Advertisements

3.4.2 Analysis Advisors

This section describes the following Advisors:

- [Get Abandonment Propensity](#)
- [Get Likely Web Action](#)
- [Get Likely Web Duration](#)
- [Get Web Support Types](#)

3.4.2.1 Get Abandonment Propensity

The Advisor Get Abandonment Propensity returns the likelihood of abandonment for a customer.

[Table 3–20](#) describes the parameters for the Advisor Get Abandonment Propensity.

Table 3–20 Advisor Get Abandonment Propensity

Parameter	Description
Advisor Name	Get Abandonment Propensity
Session Keys	Session / Current Web Interaction / Interaction Id

Table 3–20 (Cont.) Advisor Get Abandonment Propensity

Parameter	Description
Request Data	None
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Abandonment Propensity
Group Decision	Select Abandonment Propensity
Default Choices	None
Logic	None
Pre-condition	None

Table 3–21 describes the parameters for the decision for the Advisor Get Abandonment Propensity.

Table 3–21 Decision for Advisor Get Abandonment Propensity

Parameter	Description
Decision Name	Select Abandonment Propensity
Select Choices From	Abandonment
Number of Choices to Select	1
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	None

Table 3–22 describes the parameters for the choice group for the Advisor Get Abandonment Propensity.

Table 3–22 Choice Group for Advisor Get Abandonment Propensity

Parameter	Description
Choice Group Name	Abandonment (Choice Group)
Choice Attributes	Name Likelihood = Get Choice Likelihood ("AbandonmentModel", this)
Scores	Maximize Likelihood = Likelihood
Choice Events	None
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Table 3–23 describes the parameters for the model for the Advisor Get Abandonment Propensity.

Table 3–23 Model for Advisor Get Abandonment Propensity

Parameter	Description
Model Name	Abandonment Model (Choice Model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Abandonment
Mutually Exclusive	Yes
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	None
Model Name	Abandonment Model (Choice Model)

Dependencies:

- Informant - Close Session
- Function - Get Choice Likelihood (Model Name, Choice)

3.4.2.2 Get Likely Web Action

The Advisor Get Likely Web Action predicts the most likely Web Action that a particular customer will perform next.

Table 3–24 describes the parameters for the Advisor Get Likely Web Action.

Table 3–24 Advisor Get Likely Web Action

Parameter	Description
Advisor Name	Get Likely Web Action
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	None
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Likely Web Action
Group Decision	Select Likely Web Action
Default Choices	None
Logic	None
Pre-condition	None

Table 3–25 describes the parameters for the decision for the Advisor Get Likely Web Action.

Table 3–25 Decision for Advisor Get Likely Web Action

Parameter	Description
Decision Name	Select Likely Web Action
Select Choices From	Web Actions
Number of Choices to Select	1
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	None

Table 3–26 describes the parameters for the choice group for the Advisor Get Likely Web Action.

Table 3–26 Choice Group for Advisor Get Likely Web Action

Parameter	Description
Choice Group Name	Web Actions (Choice Group)
Choice Attributes	Name Likelihood = Get Choice Likelihood ("WebActionModel", this)
Scores	Maximize Likelihood = Likelihood
Choice Events	None
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Table 3–27 describes the parameters for the model for the Advisor Get Likely Web Action.

Table 3–27 Model for Advisor Get Likely Web Action

Parameter	Description
Model Name	Web Action Model (Choice Model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Web Actions
Mutually Exclusive	No
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	Customer Action
Temporary Data Storage	None
Model Name	Web Action Model (Choice Model)

Dependencies:

- Informant - Customer Action
- Function - Get Choice Likelihood (Model Name, Choice)

3.4.2.3 Get Likely Web Duration

The Advisor Get Likely Web Duration predicts the length of time that a particular customer will spend on the site.

[Table 3–28](#) describes the parameters for the Advisor Get Likely Web Duration.

Table 3–28 Advisor Get Likely Web Duration

Parameter	Description
Advisor Name	Get Likely Web Duration
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	None
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Likely Web Duration
Group Decision	Select Likely Web Duration
Default Choices	None
Logic	None
Pre-condition	None

[Table 3–29](#) describes the parameters for the decision for the Advisor Get Likely Web Duration.

Table 3–29 Decision for Advisor Get Likely Web Duration

Parameter	Description
Decision Name	Select Likely Web Duration
Select Choices From	Web Site Duration
Number of Choices to Select	1
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	None

[Table 3–30](#) describes the parameters for the choice group for the Advisor Get Likely Web Duration.

Table 3–30 Choice Group for Advisor Get Likely Web Duration

Parameter	Description
Choice Group Name	Web Site Duration (Choice Group)
Choice Attributes	Name Likelihood = Get Choice Likelihood ("WebSiteDurationModel", this)
Scores	Maximize Likelihood = Likelihood
Choice Events	None
Choice Eligibility	None
Group Attributes	None
Group Eligibility	None
Dynamic Choices	None

Table 3–31 describes the parameters for the model for the Advisor Get Likely Web Duration.

Table 3–31 Model for Advisor Get Likely Web Duration

Parameter	Description
Model Name	Web Site Duration Model (Choice Model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Web Site Duration
Mutually Exclusive	Yes
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	None
Model Name	Web Site Duration Model (Choice Model)

Dependencies:

- Informant - Customer Action
- Function - Get Choice Likelihood (Model Name, Choice)

3.4.2.4 Get Web Support Types

The Advisor Get Web Support Types returns the likelihood for a customer to accept a chat request or email support if offered.

Table 3–32 describes the parameters for the Advisor Get Web Support Types.

Table 3–32 Advisor Get Web Support Types

Parameter	Description
Advisor Name	Get Web Support Types
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	None

Table 3–32 (Cont.) Advisor Get Web Support Types

Parameter	Description
External System	Web E-Commerce
Order	0
Force session close	No
Decision	Select Web Supports
Group Decision	Select Web Supports
Default Choices	None
Logic	None
Pre-condition	None

Table 3–33 describes the parameters for the decision for the Advisor Get Web Support Types.

Table 3–33 Decision for Advisor Get Web Support Types

Parameter	Description
Decision Name	Select Web Support Types
Select Choices From	Web Support Types
Number of Choices to Select	2
Select at Random	No
Target Segments	Default
Priorities for Default Segment	Maximize Likelihood 100%
Pre Selection Logic	None
Post Selection Logic	Generate learning entries for (int i = 0; i < choices.size(); i++) {choices.get(i).recordEvent("presented");}

Table 3–34 describes the parameters for the choice group for the Advisor Get Web Support Types.

Table 3–34 Choice Group for Advisor Get Web Support Types

Parameter	Description
Choice Group Name	Web Support Types (Choice Group)
Choice Attributes	Name Likelihood of Usage = Predicted by Web Support Usage Model: Used Threshold = 0.5
Scores	Maximize Likelihood = Likelihood of Usage
Choice Events	Presented Used
Choice Eligibility	Choice / Likelihood of Usage > choice / Threshold
Group Attributes	None
Group Eligibility	None

Table 3–34 (Cont.) Choice Group for Advisor Get Web Support Types

Parameter	Description
Dynamic Choices	None

Table 3–35 describes the parameters for the model for the Advisor Get Web Support Types.

Table 3–35 Model for Advisor Get Web Support Types

Parameter	Description
Model Name	Web Support Usage Model (Choice Event Model)
Model Setting	Use for prediction, Randomize Likelihood, Default time window, Algorithm: Bayesian
Choice Group	Web Support Types
Base Event	Presented
Positive Events	Used
Partitioning Attributes	None
Excluded Attributes	None
Learn Location	On session close
Temporary Data Storage	None

Dependencies:

- Informant - Web Support Feedback

3.5 Inline Service Informants

This section describes the following Informants:

- [Initiate Session](#)
- [Identify Customer](#)
- [Page Turn](#)
- [Performed Search](#)
- [Customer Action](#)
- [Added To Cart](#)
- [Offer Response](#)
- [Web Support Feedback](#)
- [Close Session](#)

3.5.1 Initiate Session

The Informant Initiate Session creates the session for the interaction and updates current web interaction attributes for the session.

Table 3–36 describes the parameters for the Informant Initiate Session.

Table 3–36 Informant Initiate Session

Parameter	Description
Informant Name	Initiate Session
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Origin To Website => Session / Current Web Interaction / Web Interaction / Origin To Website Time Of Day => Session / Current Web Interaction / Web Interaction / Time Of Day User Location => Session / Current Web Interaction / Web Interaction / Web User Location
External System	Web E-Commerce
Order	0
Force session close	No
Logic	None
Pre-condition	None

3.5.2 Identify Customer

The Informant Identify Customer updates and triggers loading of the following customer related attributes for the session:

- Customer Profile
- Interaction History
- Purchase History
- Campaign History

The caller provides Oracle RTD with the Customer Id after a customer is identified while on the web site. In most cases, identification can be made upon user logon or by previously set cookies on the caller’s web browser.

[Table 3–37](#) describes the parameters for the Informant Identify Customer.

Table 3–37 Informant Identify Customer

Parameter	Description
Informant Name	Identify Customer
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Customer Id => Customer / Customer Id
External System	Web E-Commerce
Order	0
Force session close	No
Logic	None
Pre-condition	None

3.5.3 Page Turn

The Informant Page Turn updates current web interaction attributes (Web Pages) for the session. Identifies any given page a customer navigates to that needs to be tracked

Table 3–38 describes the parameters for the Informant Page Turn.

Table 3–38 Informant Page Turn

Parameter	Description
Informant Name	Page Turn
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Current Page => Session / Current Web Interaction / Current Page Current Page Type => Session / Current Web Interaction / Current Page Type Previous Page => Session / Current Web Interaction / Previous Page Previous Page Type => Session / Current Web Interaction / Previous Page Type Time Spent on Previous Page => Session / Current Web Interaction / Time Spent on Previous Page Current Viewed Product Id => Session / Current Web Interaction / Current Viewed Product Id
External System	Web E-Commerce
Order	0
Force session close	No
Logic	(Asynchronous) Update the nested Web Pages session attributes
Pre-condition	None

Supplied Current Viewed Product Id can be used later as the base product id in making cross sell or up sell decision.

3.5.4 Performed Search

The Informant Performed Search updates current web interaction attributes (Search Keywords) for the session. It identifies search words used by a user or customer.

Table 3–39 describes the parameters for the Informant Performed Search.

Table 3–39 Informant Performed Search

Parameter	Description
Informant Name	Performed Search
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Search Keyword
External System	Web E-Commerce
Order	0
Force session close	No
Logic	(Asynchronous) Update the nested Searches session attributes
Pre-condition	None

3.5.5 Customer Action

The Informant Customer Action updates current web interaction attributes (Web Action) for the session. It identifies key actions performed by a customer that can later be applied to models, for example, Update Customer Profile, Register, Un-register, Chat request.

[Table 3–40](#) describes the parameters for the Informant Customer Action.

Table 3–40 Informant Customer Action

Parameter	Description
Informant Name	Customer Action
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Action Name
External System	Web E-Commerce
Order	0
Force session close	No
Logic	Update Web Action Model (Choice Model) Create or Update Web Action session variable of current interaction
Pre-condition	None

Set Action Name as a choice name into the Web Action Model.

3.5.6 Added To Cart

The Informant Added To Cart updates current web interaction attributes (Cart Item) for the session. It registers a shopping cart item along with its quantity added during the session.

[Table 3–41](#) describes the parameters for the Informant Added To Cart.

Table 3–41 Informant Added to Cart

Parameter	Description
Informant Name	Customer Action
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Product Id Quantity
External System	Web E-Commerce
Order	0
Force session close	No
Logic	Update Web Action Model (Choice Model) Create or Update Shopping Cart Item session variable of current interaction
Pre-condition	None

3.5.7 Offer Response

The Informant Offer Response creates prediction model entries and also captures shopping cart addition events. It identifies if an offer (Up Sell, Cross Sell, Promotion, Ad) is clicked, added to cart, or purchased by a web user or customer.

[Table 3–42](#) describes the parameters for the Informant Offer Response.

Table 3–42 Informant Offer Response

Parameter	Description
Informant Name	Offer Response
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Offer Id {choice id} Offer Type {Up Sell, Cross Sell, Promotion, Ad} Event {Interested, Added To Cart, Purchased}
External System	Web E-Commerce
Order	0
Force session close	No
Logic	If event = Interested then if offer type = Promotion then create/update Clicked Promotion (increase count if previously exist) if offer type = Ad then create/update Clicked Ad (increase count if previously exist) Record the event into the appropriate Model for the choice (determined by choice id / offer id)
Pre-condition	None

3.5.8 Web Support Feedback

The Informant Web Support Feedback updates the Web Support prediction model.

[Table 3–43](#) describes the parameters for the Informant Web Support Feedback.

Table 3–43 Informant Web Support Feedback

Parameter	Description
Informant Name	Web Support Feedback
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Choice Name Choice Event
External System	Web E-Commerce
Order	0
Force session close	No
Logic	If Choice Event = Used then record the event into the Web Support Usage Model
Pre-condition	None

Dependencies:

- Function - Set Choice Event Model (Model Name, Choice Name, Choice Event)

3.5.9 Close Session

The Informant Close Session creates analytical prediction model entries, triggers learning, and formally closes out the Oracle RTD session.

[Table 3–44](#) describes the parameters for the Informant Close Session.

Table 3–44 Informant Close Session

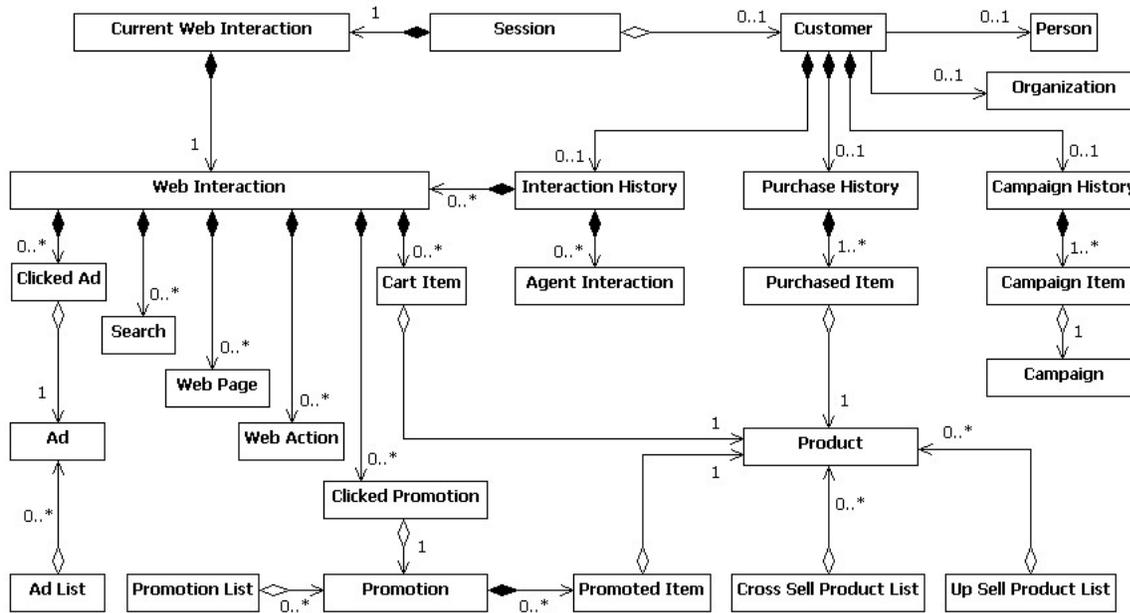
Parameter	Description
Informant Name	Close Session
Session Keys	Session / Current Web Interaction / Interaction Id
Request Data	Abandonment Flag Interaction Duration
External System	Web E-Commerce
Order	0
Force session close	No
Logic	Update Abandonment Model. Possible values for Abandonment Flag are: "Abandoned" "Not Abandoned" Update Web Site Duration Model. Possible values for Interaction Duration are: "00 to 05 Minutes" "05 to 10 Minutes" "10 to 15 Minutes" "15 to 20 Minutes" "20 Minutes or Greater"
Pre-condition	None

Dependencies:

- Function - Set Choice Model (Model Name, Choice Name)

3.6 Inline Service Entities

The following diagram shows the relationships between the Entities defined in the Base E-Commerce Inline Service.



Notation

The diagram shows the standard notation used in UML class diagrams, with each directed line representing a relationship *from* element A *to* element B, as follows:

- A line with no diamond represents an **association** between elements A and B.
- A line with a clear diamond represents an **aggregation**, with element A "owning" element B. However, with aggregation, removing an instance of element A does not remove the corresponding B element instances.
- A line with a filled diamond represents a **composition**, with element A "owning" element B. For a composition, removing an instance of element A also removes the corresponding B element instances.

The multiplicity of a relationship restricts how many element B instances the relationship may have. The restriction denotes either a precise limit, such as 1 or 0..1, or an open-ended upper limit, such as "zero or more" or "one or more."

For example:

- A Web Interaction may have many Web Pages. Deleting a particular Web Interaction automatically deletes the corresponding Web Pages.
- A Cart Item has one Product. Deleting the Cart Item does not delete the corresponding Product.

System-Oriented Entities

The Session Entity is a built-in Oracle RTD Entity for maintaining session attribute values.

The following Entities are used for dynamic choice retrieval from external data sources: Ad List, Promotion List, Cross Sell Product List, and Up Sell Product List.

Entities Outline

The Current Web Interaction Entity is a session attribute that keeps track of the current user interaction with the client system.

The Current Web Interaction Entity references the Web Interaction Entity, that itself keeps track of the following data:

- Clicked advertisements, using the Clicked Ad Entity
- Clicked promotions, using the Clicked Promotion Entity
- Searches performed, using the Search Entity
- Actions performed, using the Web Action Entity
- Visited Web pages, using the Web Page Entity
- Items added to the shopping cart, using the Cart Item Entity

The Customer Entity is a session attribute that contains details of the customer profile as well as past customer behavior.

A customer can be a Person or an Organization.

The Customer may have historical information, in Purchase History, Campaign History, and Interaction History. Past customer interactions could be either or both of the following:

- Self-service Web-based interactions, using the Web Interaction Entity
- Interactions through agents, using the Agent Interaction Entity

After a Customer Id has been identified and supplied to Oracle RTD by the Identify Customer Informant, Oracle RTD retrieves data from external data sources for the Customer Entity and for its associated entity attributes.

In addition to the Entity attributes that are normally mapped to a data source directly, derived attributes are also included, which obtain their values via Java functions that utilize the applicable raw data extracted from data sources as inputs.

This section describes the following Entities:

- [Session Entity](#)
- [Ad Entity](#)
- [Ad List Entity](#)
- [Agent Interaction Entity](#)
- [Campaign Entity](#)
- [Campaign History Entity](#)
- [Campaign Item Entity](#)
- [Cart Item Entity](#)
- [Clicked Ad Entity](#)
- [Clicked Promotion Entity](#)
- [Cross Sell Product List Entity](#)
- [Current Web Interaction Entity](#)
- [Customer Entity](#)
- [Interaction History Entity](#)

- Organization Entity
- Person Entity
- Product Entity
- Promoted Item Entity
- Promotion Entity
- Promotion List Entity
- Purchased Item Entity
- Purchase History Entity
- Rank Offers Entity
- Search Entity
- Up Sell Product List Entity
- Web Action Entity
- Web Interaction Entity
- Web Page Entity

3.6.1 Session Entity

(Key = Web Session Id)

Table 3–45 Session Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Customer	No	Customer	Yes	Yes	None
Current Web Interaction	No	Current Web Interaction	Yes	Yes	None
Rank Offers	No	Rank Offers	No	No	Utilized when ranking offers directly from an Offer Advisor instead of read from a data source.
Supplied Product Id	No	String	No	No	Utilized when ranking offers directly from an Offer Advisor instead of read from a data source.

3.6.2 Ad Entity

The Ad Entity is used in conjunction with the Ad List Entity and is used for the dynamic choice associated to the Ad Choice group.

Table 3–46 Ad Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Ad Id	No	String	No	No	Key
Category	No	String	No	No	None
Name	No	String	No	No	None
Type	No	String	No	No	None

3.6.3 Ad List Entity

The Ad List Entity is used in conjunction with the Ad Entity and is used for the dynamic choice associated to the Ad Choice group.

Table 3–47 Ad List Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Active	No	String	No	No	Key
Ads	Yes	Ad	No	No	The Ads attribute is an array attribute based on the Ad Entity.

3.6.4 Agent Interaction Entity

The Agent Interaction Entity contains attributes related to agent interactions that have taken place with a customer. This Entity is used to create an array of agent interactions within the Interaction History Entity, which in turn is associated to the Customer Entity.

Table 3–48 Agent Interaction Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Interaction Id	No	String	No	No	None
Agent Id	No	String	No	No	None
Agent Location	No	String	No	No	None
Customer Location	No	String	No	No	None
Interaction Channel	No	String	No	No	None
Interaction Date	No	None	No	No	None
Interaction Duration	No	Integer	No	No	None
Interaction Reason	No	String	No	No	None
Interaction Status	No	String	No	No	None
Interaction Type	No	String	No	No	None
Time Of Day	No	String	No	No	None

3.6.5 Campaign Entity

The Campaign Entity contains attributes related to campaigns that have been associated to a customer. This Entity is associated to the customer via the Campaign Item and Campaign History entities.

Table 3–49 Campaign Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Campaign Id	No	String	No	No	None
Category	No	String	Yes	Yes	None
Name	No	String	Yes	Yes	None
Period	No	String	Yes	Yes	None
Type	No	String	Yes	Yes	None

3.6.6 Campaign History Entity

The Campaign History Entity contains attributes related to campaigns that have been associated to a customer. This Entity is associated to the Customer Entity and also contains the Campaign Items Entity.

Table 3–50 Campaign History Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Last Campaign Category	No	String	Yes	Yes	Default Value based on the Get Last Campaign function
Last Campaign Date	No	Date	Yes	Yes	Default Value based on the Get Last Campaign Date function
Last Campaign Delivery Method	No	String	Yes	Yes	Default Value based on the Get Last Campaign Delivery Method function
Last Campaign Name	No	String	Yes	Yes	Default Value based on the Get Last Campaign Name function
Last Campaign Type	No	String	Yes	Yes	Default Value based on the Get Last Campaign Type function
Days Since Last Campaign	No	Integer	Yes	Yes	Default Value based on the Get Days Since Last Campaign

Table 3–50 (Cont.) Campaign History Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Campaign Items	Yes	Campaign Item	Yes	Yes	The Campaign Items attribute is an array attribute based on the Campaign Item Entity.

3.6.7 Campaign Item Entity

The Campaign Item Entity contains attributes related to campaigns that have been associated to a customer. This Entity is associated to the customer via the Campaign History Entity and contains the Campaign Entity.

Table 3–51 Campaign Item Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Campaign Date	No	Date	Yes	Yes	None
Delivery Method	No	String	Yes	Yes	None
Campaign	No	Campaign	Yes	Yes	The Campaign attribute is based on the Campaign Entity.

3.6.8 Cart Item Entity

The Cart Item Entity contains attributes related to the products that a customer has put in their shopping cart in their web session. The Cart Item Entity contains the Product Entity, and is itself embedded in the Web Interaction Entity.

Table 3–52 Cart Item Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Quantity	No	Integer	Yes	Yes	None
Product	No	Product	Yes	Yes	None

3.6.9 Clicked Ad Entity

The Clicked Ad Entity contains attributes related to the ads which a customer may have clicked during their web session. The Clicked Ad Entity contains the Ad Entity, and is itself embedded in the Web Interaction Entity.

Table 3–53 Clicked Ad Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Count	No	Integer	Yes	Yes	None
Ad	No	Ad	Yes	Yes	None

3.6.10 Clicked Promotion Entity

The Clicked Promotion Entity contains attributes related to the ads which a customer may have clicked during their web session. The Clicked Promotion Entity contains the Promotion Entity, and is itself embedded in the Web Interaction Entity.

Table 3–54 Clicked Promotion Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Count	No	Integer	Yes	Yes	None
Promotion	No	Promotion	Yes	Yes	None

3.6.11 Cross Sell Product List Entity

The Cross Sell Product List Entity is used in conjunction with the Products Entity (instantiated as Cross Sell Products), and is used for the dynamic choice associated to the Cross Sell Offers Choice group.

Table 3–55 Cross Sell Product List Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Product Id	No	String	No	No	None
Cross Sell Products	Yes	Product	No	No	The Cross Sell Products attribute is an array attribute based on the Product Entity.
Cross Sell Products - Product Id	No	String	No	No	None

3.6.12 Current Web Interaction Entity

The Current Web Interaction Entity contains attributes related to what a customer is doing during their current web session. The Current Web Interaction Entity contains its own attributes as well as attributes from the Web Interaction Entity.

See [Section 3.6.27.1, "Derivation of Web Interaction Attributes in Referencing Entities"](#) for details of how the Web Interaction attributes Interaction Date, Start Time, and Total Duration in Minutes are derived.

Table 3–56 Current Web Interaction Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Interaction Id	No	String	Yes	Yes	None
Current Page	No	String	Yes	Yes	None
Current Page Type	No	String	Yes	Yes	None
Current Viewed Product Id	No	String	Yes	Yes	None
Last Added Product Id	No	String	Yes	Yes	None
Previous Page	No	String	Yes	Yes	None
Previous Page Type	No	String	Yes	Yes	None

Table 3–56 (Cont.) Current Web Interaction Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Time Spent on Previous Page	No	Integer	Yes	Yes	None
Web Interaction	No	Web Interaction	Yes	Yes	None

3.6.13 Customer Entity

The Customer Entity contains attributes related to the profile of the customer. The Customer Entity contains its own attributes and links in the Campaign History, Interaction History, Organization, Person, and Purchase History Entities.

Table 3–57 Customer Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Customer Id	No	String	No	No	Key
Address City	No	String	Yes	Yes	None
Address Country	No	String	Yes	Yes	None
Address Postal Code	No	String	Yes	Yes	None
Address Region	No	String	Yes	Yes	None
Address State Province	No	String	Yes	Yes	None
Credit Hold	No	String	Yes	Yes	None
Life Time Value Score	No	Double	Yes	Yes	None
Offline Churn Propensity	No	Double	Yes	Yes	None
Phone Area Code	No	String	Yes	Yes	None
Preferred Language	No	String	Yes	Yes	None
Start Date	No	Date	Yes	Yes	None
Status	No	String	Yes	Yes	None
Target Market Segment	No	String	Yes	Yes	None
Tenure	No	Integer	Yes	Yes	None
Total Credit Limit	No	Double	Yes	Yes	None
Type	No	String	Yes	Yes	Default Value determined by Get Customer Type function.
Campaign History	No	Campaign History	Yes	Yes	The Campaign History attribute is based on the Campaign History Entity. See Campaign History Entity for attribute breakdown.

Table 3–57 (Cont.) Customer Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Interaction History	No	Interaction History	Yes	Yes	The Interaction History attribute is based on the Interaction History Entity. See Interaction History Entity for attribute breakdown.
Organization	No	Organization	Yes	Yes	The Organization attribute is based on the Organization Entity. See Organization Entity for attribute breakdown.
Person	No	Person	Yes	Yes	The Person attribute is based on the Person Entity. See Person Entity for attribute breakdown.
Purchase History	No	Purchase History	Yes	Yes	The Purchase History attribute is based on the Purchase History Entity. See Purchase History Entity for attribute breakdown.

3.6.14 Interaction History Entity

The Interaction History Entity contains attributes that record what a customer has done in the past regarding previous interactions. The Interaction History Entity contains derived attributes from both previous Agent Interactions and previous Web Interactions.

See [Section 3.6.27.1, "Derivation of Web Interaction Attributes in Referencing Entities"](#) for details of how the Past Web Interactions attributes Interaction Date, Start Time, and Total Duration in Minutes are derived.

Table 3–58 Interaction History Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Agent Interaction Reasons In Past 30 Days	Yes	String	Yes	Yes	Default determined by Get Agent Interaction Reasons In Past Days function

Table 3–58 (Cont.) Interaction History Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Agent Interaction Types in Past 30 Days	Yes	String	Yes	Yes	Default determined by Get Agent Interaction Types In Past Days function
Days Since Last Agent Interaction	No	Integer	Yes	Yes	Default determined by Get Days Since Last Agent Interaction function
Days Since Last Interaction	No	Integer	Yes	Yes	Default determined by Maximum function
Days Since Last Web Interaction	No	Integer	Yes	Yes	Default determined by Get Days Since Last Web Interaction function
Interaction Types In Past 30 Days	No	String	Yes	Yes	Default determined by Get Interaction Types function
Last Agent Interaction Status	No	String	Yes	Yes	Default determined by Get Last Agent Interaction Status function
Last Agent Interaction Type	No	String	Yes	Yes	Default determined by Get Last Agent Interaction Type function
Last Interaction Type	No	String	Yes	Yes	Default determined by Get Last Interaction Type function
Number of Agent Interactions In Past 30 Days	No	Integer	Yes	Yes	Default determined by Get Number Of Agent Interaction in Past Days function
Number of Agent Interactions In Past 90 Days	No	Integer	Yes	Yes	Default determined by Get Number Of Agent Interaction in Past Days function
Number of Agent Interactions In Past Year	No	Integer	Yes	Yes	Default determined by Get Number Of Agent Interaction in Past Days function
Number of Web Interactions In Past 30 Days	No	Integer	Yes	Yes	Default determined by Get Number Of Web Interaction in Past Days function
Number of Web Interactions In Past 90 Days	No	Integer	Yes	Yes	Default determined by Get Number Of Web Interaction in Past Days function
Number of Web Interactions In Past Year	No	Integer	Yes	Yes	Default determined by Get Number Of Web Interaction in Past Days function

Table 3–58 (Cont.) Interaction History Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Performed Web Actions In Past 30 Days	Yes	String	Yes	Yes	Default determined by Get Performed Web Actions in Past Days function
Past Agent Interactions	Yes	Agent Interaction	No	No	None
Past Web Interactions	Yes	Web Interaction	No	No	None

3.6.15 Organization Entity

The Organization Entity contains attributes related to the profile of an Organization. The Organization Entity is linked to the session through the Customer Entity.

Table 3–59 Organization Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Annual Gross Profit	No	Double	Yes	Yes	None
Annual Revenue	No	Double	Yes	Yes	None
Business Partner Flag	No	String	Yes	Yes	None
Established Service	No	Integer	Yes	Yes	None
Line Of Business	No	String	Yes	Yes	None
Number of Employees	No	Integer	Yes	Yes	None
Number Of Years Established	No	Integer	Yes	Yes	None
Size	No	String	Yes	Yes	None
Type	No	String	Yes	Yes	None

3.6.16 Person Entity

The Person Entity contains attributes related to the profile of a Person. The Person Entity is linked to the session through the Customer Entity.

Table 3–60 Person Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Age	No	Integer	Yes	Yes	None
Annual Income	No	Double	Yes	Yes	None
Credit Score	No	Integer	Yes	Yes	None
Education Level	No	String	Yes	Yes	None
Ethnicity	No	String	Yes	Yes	None
Gender	No	String	Yes	Yes	None
Marital Status	No	String	Yes	Yes	None
Net Worth	No	Double	Yes	Yes	None

Table 3–60 (Cont.) Person Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Number Of Children	No	Integer	Yes	Yes	None
Profession	No	String	Yes	Yes	None

3.6.17 Product Entity

The Product Entity contains attributes related to a generic Product. The Product Entity is used as a reference entity under the Cart Item, Cross Sell Product List, Promoted Item, Purchased Item, and Up Sell Product Entities.

Table 3–61 Product Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Product Id	No	String	No	No	Key
Category	No	String	Yes	Yes	None
Name	No	String	Yes	Yes	None
Popularity Rank	No	Integer	Yes	Yes	None
Product Line	No	String	Yes	Yes	None
Type	No	String	Yes	Yes	None
Unit Price	No	Double	Yes	Yes	None

3.6.18 Promoted Item Entity

The Promoted Item Entity contains attributes related to a Promoted Item. The Promoted Item Entity is used in conjunction with the Promotion Entity, which can contain multiple promoted Items.

Table 3–62 Promoted Item Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Discount Rate	No	Double	Yes	Yes	None
Promoted Product	No	Product	Yes	Yes	None

3.6.19 Promotion Entity

The Promotion Entity is used in conjunction with the Promotion List Entity, and is used for the dynamic choice associated to the Promotions Choice group.

Table 3–63 Promotion Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Promotion Id	No	String	No	No	None
Category	No	String	No	No	None
Days Left	No	Integer	No	No	Default Value set by Get Days Left function

Table 3–63 (Cont.) Promotion Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Duration In Days	No	Integer	No	No	Default Value set by Get Duration In Days function
Effective Date	No	Date	No	No	None
Expiry Date	No	Date	No	No	None
Name	No	String	No	No	None
Period	No	String	No	No	None
Type	No	String	No	No	None
Promoted Items	Yes	Promoted Item	No	No	The Promoted Items attribute is based on the Promoted Item Entity. See Promoted Item Entity for attribute breakdown.

3.6.20 Promotion List Entity

The Promotion List Entity is used in conjunction with the Promotion Entity, and is used for the dynamic choice associated to the Promotions Choice group.

Table 3–64 Promotion List Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Active	No	String	No	No	Key
Promotions	Yes	Promotion	No	No	The Promotions attribute is based on the Promotion Entity. See Promotion Entity for attribute breakdown.

3.6.21 Purchased Item Entity

The Purchased Item Entity contains attributes related to a purchased item. This Entity is associated to the Customer Entity through the Purchase History Entity.

Table 3–65 Purchased Item Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Purchase Amount	No	Double	Yes	Yes	None
Purchase Date	No	Date	Yes	Yes	None
Purchased Product	No	Product	Yes	Yes	None

3.6.22 Purchase History Entity

The Purchased History Entity contains attributes related to a customer's past purchases. This Entity is associated to the session via the Customer Entity.

Table 3–66 Purchase History Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Days Since Last Purchase	No	Integer	Yes	Yes	Default Value set by Get Days Since Last Purchase function
Last Purchase Amount	No	Double	Yes	Yes	Default Value set by Get Last Purchase Amount function
Last Purchased Product	No	String	Yes	Yes	Default Value set by Get Last Purchased Product function
Last Purchased Product Line	No	String	Yes	Yes	Default Value set by Get Last Purchased Product Line function
Product Lines Owned	Yes	String	Yes	Yes	Default Value set by Get Product Lines Owned function
Total Amount Spent	No	Double	Yes	Yes	Default Value set by Get Total Amount Spent function
Total Amount Spent in Last 90 Days	No	Double	Yes	Yes	Default Value set by Get Total Amount Spent In Last 90 Days function
Purchased Items	Yes	Purchased Item	Yes	Yes	The Purchased Items array attribute is based on the Purchased Item Entity. See Purchased Item Entity for attribute breakdown.

3.6.23 Rank Offers Entity

The Rank Offers Entity is used to store arrays of different offer types that are passed to the Inline Service via an Advisor. After it is filled by the Advisor input, the Entity is then used to populate the corresponding dynamic choice.

Table 3–67 Rank Offers Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Ads	Yes	String	No	No	None
CrossSellOffers	Yes	String	No	No	None
In Use	No	Boolean	No	No	Default Value is False
Promotions	Yes	String	No	No	
UpSellOffers	Yes	String	No	No	

3.6.24 Search Entity

The Search Entity is used to store attributes related to search strings performed by a customer. This Entity is used by the Web Interaction Entity as an array attribute.

Table 3–68 Search Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Count	No	Integer	Yes	Yes	None
Keyword	No	String	Yes	Yes	None

3.6.25 Up Sell Product List Entity

The Up Sell Product List Entity is used in conjunction with the Products Entity (instantiated as Up Sell Products), and is used for the dynamic choice associated to the Up Sell Offers Choice group.

Table 3–69 Up Sell Product List Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Product Id	No	String	No	No	None
Up Sell Products	Yes	Product	No	No	The Up Sell Products attribute is an array attribute based on the Product Entity.

3.6.26 Web Action Entity

The Web Action Entity is used to store attributes related to actions performed by a customer while on the web. This Entity is used by the Web Interaction Entity as an array attribute.

Table 3–70 Web Action Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Name	No	String	No	No	Key
Category	No	String	Yes	Yes	None
Type	No	String	Yes	Yes	None
Web Action Id	No	String	Yes	Yes	None

3.6.27 Web Interaction Entity

The Web Interaction Entity contains attributes related to a generic Web Interaction. This Entity is used as a reference entity under for the Current Web Interaction Entity as well as an array attribute (Past Web Interactions) under the Interaction History Entity.

See [Section 3.6.27.1, "Derivation of Web Interaction Attributes in Referencing Entities"](#) for details of how the Interaction Date, Start Time, and Total Duration in Minutes attributes are used in the Current Web Interaction Entity and the Interaction History Entity.

Table 3–71 Web Interaction Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Interaction Id	No	String	No	No	Key
Clicked Ads	Yes	Clicked Ad	Yes	Yes	The Clicked Ads attribute is an array attribute based on the Clicked Ad Entity. See the Clicked Ad Entity for attribute breakdown.

Table 3–71 (Cont.) Web Interaction Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Clicked Promotions	Yes	Clicked Promotion	Yes	Yes	The Clicked Promotions attribute is an array attribute based on the Clicked Promotion Entity. See the Clicked Promotion entity for attribute breakdown.
Interaction Date	No	Date	Yes	Yes	None
Origin To Website	No	String	Yes	Yes	None
Performed Actions	Yes	Web Action	Yes	Yes	The Performed Actions attribute is an array attribute based on the Web Action Entity. See the Web Action Entity for attribute breakdown.
Performed Searches	Yes	Search	Yes	Yes	The Performed Searches attribute is an array attribute based on the Search Entity. See the Search Entity for attribute breakdown.
Shopping Cart Items	Yes	Cart Item	Yes	Yes	The Shopping Cart Items attribute is an array attribute based on the Cart Item Entity. See the Cart Item Entity for attribute breakdown.
Start Time	No	Date	No	No	Attribute used to derive the Web Interaction attribute Total Duration in Minutes when used in the Current Web Interaction Entity.
Time Of Day	No	String	Yes	Yes	None
Total Duration in Minutes	No	String	Yes	Yes	None
Visited Pages	Yes	Web Page	Yes	Yes	The Visited Pages attribute is an array attribute based on the Web Page Entity. See the Web Page Entity for attribute breakdown.
Web User Location	No	String	Yes	Yes	None

3.6.27.1 Derivation of Web Interaction Attributes in Referencing Entities

When the [Current Web Interaction Entity](#) and the [Interaction History Entity](#) reference the Web Interaction Entity, special considerations apply to the derivation of certain attributes, as follows:

Table 3–72 Derivation of Web Interaction Attributes in Referencing Entities

Web Interaction Attribute	As used in the Current Web Interaction Entity attribute Web Interaction	As used in the Interaction History Entity attribute Past Web Interactions
Interaction Date	NA	Retrieved from the external data source using Web Interaction mapping

Table 3–72 (Cont.) Derivation of Web Interaction Attributes in Referencing Entities

Web Interaction Attribute	As used in the Current Web Interaction Entity attribute Web Interaction	As used in the Interaction History Entity attribute Past Web Interactions
Start Date	Initialized by Informant Initiate Session with the Current Time	Not initialized
Total Duration in Minutes	Computed as the difference between Web Interaction-Start Time and Current Time	Retrieved from the external data source using Web Interaction mapping

3.6.28 Web Page Entity

The Web Page Entity is used to store attributes related to the attributes of a web page that a customer has visited while on the web. This Entity is used by the Web Interaction Entity as an array attribute.

Table 3–73 Web Page Entity

Attribute	Array	Type	Show In DC	Use for Analysis	Comments
Web Page Id	No	String	No	No	Key
Average Time Spent On Page	No	Double	Yes	Yes	None
Category	No	String	Yes	Yes	None
Count	No	Integer	Yes	Yes	None
Name	No	String	Yes	Yes	None
Type	No	String	Yes	Yes	None

3.7 Inline Service Functions

This section describes the functions used in the Base E-Commerce Inline Service.

Table 3–74 Base E-Commerce Inline Service Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Get Ad List	None	Advertisements (Array of Ad)	Used for Advertisements dynamic choices. See corresponding functions: Set Session Rank Offers.	This function is used to return array of Ad Entity for Advertisement dynamic choice. The result is either ranked advertisements array supplied as incoming parameter of Advisors or array of advertisements loaded using data source mapping.
Get Agent Interaction Reasons In Past Days	Past Agent Interactions (Array of Agent Interaction), Days (Integer)	Interaction Reasons (Array of String)	Used for deriving values for Interaction History Entity.	This function returns array of interaction reason in past given days.
Get Agent Interaction Types In Past Days	Past Agent Interactions (Array of Agent Interaction), Days (Integer)	Interaction Types (Array of String)	Used for deriving values for Interaction History Entity.	This function returns array of interaction types in past given days.

Table 3–74 (Cont.) Base E-Commerce Inline Service Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Get Campaign Names In Past Days	Campaign Items (Array of Campaign Item), Days (Integer)	Campaign Names (Array of String)	Used for deriving values for Campaign History Entity	This function returns array of campaign names delivered in past given days.
Get Choice Likelihood	Model Name (String), Choice (Choice)	Likelihood (Double)	Used for scoring in Abandonment, Web Actions, Web Site Duration choice groups	This function returns likelihood for a given inputted choice that is part of a Choice Model (as opposed to a choice even model). As inputs, the user must pass in a choice and the model name that choice is part of.
Get Cross Sell Product List	Product Id (String) to be used as base product for cross sell	Cross Sell Products (Array of Product)	Used for Cross Sell dynamic choices. See corresponding functions: Set Session Rank Offers.	This function is used to return array of Product Entity for Cross Sell dynamic choice. The result is either ranked products array supplied as incoming parameter of Advisors or array of cross sell products loaded using data source mapping.
Get Customer Type	Person (Person), Organization (Organization)	Customer Type (String)	Used by Customer Entity.	This function is used to determine customer type.
Get Days Left	Expiration Date (Date)	Number of Days Left (Integer)	Used by Promotion Entity.	This function calculates the number of days from now to expiryDate.
Get Days Since Last Agent Interaction	Past Agent Interactions (Array of Agent Interaction)	Number of Days (Integer)	Used by Interaction History Entity.	This function calculates the number of days from last agent interaction to now.
Get Days Since Last Campaign	Campaign Items (Array of Campaign Item)	Number of Days (Integer)	Used by Campaign History Entity.	This function calculates the number of days from last campaign to now.
Get Days Since Last Purchase	Purchased Items (Array of Purchased Item)	Number of Days (Integer)	Used by Purchase History Entity.	This function calculates the number of days from last purchase to now.
Get Days Since Last Web Interaction	Past Web Interactions (Array of Web Interaction)	Number of Days (Integer)	Used by Interaction History Entity.	This function calculates the number of days from last web interaction to now.
Get Duration In Days	Start Date (Date), End Date (Date)	Number of Days (Integer)	Used by Promotion Entity and various other functions.	This function calculates the number of days from start date to end date.
Get Duration In Minutes	Start Time (Date), End Time (Date)	Number of Minutes (Integer)	NA.	This function calculates the number of minutes from start time to end time.
Get Gender	Male (Boolean)	Gender In Text (String)	Used by Customer Entity.	This function determines the gender of a person. This converts the given gender type from boolean to string. In Male case, genderMale is boolean value=true fetched from data source.

Table 3–74 (Cont.) Base E-Commerce Inline Service Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Get Interaction Types In Past Days	Past Agent Interactions (Array of Agent Interaction), Past Web Interactions (Array of Web Interaction), Days (Integer)	Interaction Type in Text (String)	Used by Interaction History Entity.	This function returns interaction type in past given days.
Get Last Agent Interaction Status	Past Agent Interactions (Array of Agent Interaction)	Last Status (String)	Used by Interaction History Entity	This function returns last agent interaction status.
Get Last Agent Interaction Type	Past Agent Interactions (Array of Agent Interaction)	Last Type (String)	Used by Interaction History Entity	This function returns last agent interaction type.
Get Last Campaign Category	Campaign Items (Array of Campaign Item)	Category (String)	Used by Campaign History Entity.	This function returns last campaign category.
Get Last Campaign Delivery Method	Campaign Items (Array of Campaign Item)	Delivery Method (String)	Used by Campaign History Entity.	This function returns last campaign delivery method.
Get Last Campaign Name	Campaign Items (Array of Campaign Item)	Name (String)	Used by Campaign History Entity.	This function returns last campaign name.
Get Last Campaign Type	Campaign Items (Array of Campaign Item)	Type (String)	Used by Campaign History Entity.	This function returns last campaign type.
Get Last Interaction Type	Days Since Last Agent Interaction (Integer), Days Since Last Web Interaction (Integer)	Interaction Type (String)	Used by Interaction History Entity.	This function returns last interaction type. Return value will be Agent, Web, or Both.
Get Last Purchase Amount	Purchased Items (Array of Purchased Item)	Purchase Amount (Double)	Used by Purchase History Entity.	This function returns last purchase amount.
Get Last Purchased Product	Purchased Items (Array of Purchased Item)	Product Name (String)	Used by Purchase History Entity.	This function returns last purchased product name.
Get Last Purchased Product Line	Purchased Items (Array of Purchased Item)	Product Line (String)	Used by Purchase History Entity.	This function returns last purchased product line.
Get Number Of Agent Interaction In Past Days	Past Agent Interactions (Array of Agent Interaction), Days (Integer)	Count (Integer)	Used by Interaction History Entity.	This function counts number of agent interactions in past given days.

Table 3–74 (Cont.) Base E-Commerce Inline Service Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Get Number Of Web Interaction In Past Days	Past Web Interactions (Array of Web Interaction), Days (Integer)	Count (Integer)	Used by Interaction History Entity.	This function counts number of web interactions in past given days.
Get Performed Web Action Names In Past Days	Past Web Interactions (Array of Web Interaction), Days (Integer)	Action Names (Array of String)	Used by Interaction History Entity.	This function returns array of web action names performed in past given days.
Get Product Lines Owned	Purchased Items (Array of Purchased Item)	Product Lines (Array of String)	Used by Purchase History Entity.	This function returns product lines of customer owned.
Get Promotion List	None	Promotions (Array of Promotion)	Used for Promotion dynamic choices. See corresponding functions: Set Session Rank Offers.	This function is used to return array of Promotion Entity for Promotion dynamic choice. The result is either ranked promotion array supplied as incoming parameter of Advisors or array of promotions loaded using data source mapping.
Get Specific Choice Likelihood	Model Name (String), Choice Name (String)	Likelihood (Double)	None. See similar function: Get Choice Likelihood	This function returns likelihood for a given inputted choice that is part of a Choice Model (as opposed to a choice even model). As inputs, the user must pass in the value for the Name attribute assigned to the choice and the model name that choice is part of.
Get Total Amount Spent	Purchased Items (Array of Purchased Item)	Amount (Double)	Used by Purchase History Entity.	This function sums up the amount of customer spent so far.
Get Total Amount Spent in Last 90 Days	Purchased Items (Array of Purchased Item)	Amount (Double)	Used by Purchase History Entity.	This function sums up the amount of customer spent in past 90 days.
Get Total Duration In Minutes	Start Time (Date)	Number of Minutes (Integer)	Used by Web Interaction Entity.	This function calculates number of minutes between the given start time and now.
Get Up Sell Product List	Product Id (String) to be used as base product for up sell	Up Sell Products (Array of Product)	Used for Up Sell dynamic choices. See corresponding functions: Set Session Rank Offers.	This function is used to return array of Product Entity for Up Sell dynamic choice. The result is either ranked products array supplied as incoming parameter of Advisors or array of up sell products loaded using data source mapping.
Get Year	Date (Date)	Year (Integer)	Used by Customer Entity.	This function returns the year part of Date.
Is Web Support Type Eligible	Web Support Types Choice (Web Support Types Choice)	Eligible (Boolean)	Used by Web Support Types Choice Group.	This function returns the eligibility of a Web Support Types Choice.
Maximum	X (Integer), Y (Integer)	Maximum (Integer)	Used by Interaction History Entity.	This function returns the greater parameter.
Minutes To Now	Start Time (Date)	Minutes (Integer)	Used by Total Duration In Minutes function	This function returns time between given time in the past and now.
Multiply	a (Double, b (Double)	Result (Double)	Used by various RTD elements	This function multiplies the given parameters.
Property Reflect	NA	NA	NA	This function is used for testing. This dumps the values of session attributes.

Table 3–74 (Cont.) Base E-Commerce Inline Service Functions

Function	Inputs	Outputs	Area Utilized In	Comments
Select Product Id	None	Product Id (String)	Used as base product for Up Sell and Cross Sell dynamic choices.	This function determines what Product Id to be used as base Product Id for Up Sell or Cross Sell.
Set Choice Event Model	Choice Event Model Name (String), Choice Name (String), Choice Event (String)	None	Used by Web Support Feedback informant.	This function is used to store all of the logic needed to set the Choice Models tied to the Choice Groups.
Set Choice Model	Choice Model Name (String), Choice Name (String)	None	Used by Customer Action and Close Session informants.	This function is used to store all of the logic needed to set the Choice Models tied to the Choice Groups.
Set Session Rank Offers	Offers (Array of String)	None	Used by all offer-oriented advisors to set the optionally supplied string array of keys into session for later use as dynamic choice source. See corresponding functions: Get Ad List, Get Promotion List, Get Cross Sell Product List, Get Up Sell Product List.	This procedure is used to parse and set Rank Offers before decision selection.
Years To Now	Start Time (Date)	Years (Integer)	Used by Customer Entity.	This function calculates the number of years from given date to now.

Configuring the Base E-Commerce Inline Service

This chapter outlines the steps that you can take to configure the Base E-Commerce Inline Service.

As outlined in [Chapter 3, "Base E-Commerce Inline Service Elements,"](#) the Base E-Commerce Inline Service contains a template of metadata. This template enables customers to tie their E-Commerce front end to a variety of Oracle RTD Integration Points. Oracle RTD can use these Integration Points both for Oracle RTD decisioning and for analysis of the customer web site.

While the Inline Service contains a variety of decisioning and data transformation logic, users must still take this Inline Service and identify what elements are necessary to support their business workflow and requirements.

To achieve this alignment, Oracle recommends the following high-level steps:

1. Align business workflow with Oracle RTD Integration Points and metadata.
 - a. Identify and sequence the required Integration Points.
 - b. Identify incoming real-time data.
2. Integrate Oracle RTD with the customer front end.
 - a. Refer to the manual *Oracle Real-Time Decisions Integration with Oracle RTD* for integration methods to external applications.
3. Map entity attributes to physical data sources.
 - a. Add additional entity attributes as required by the business process.
4. Add additional logic as required.
 - a. Areas include eligibility rules, additional physical and derived attributes, performance goals and scoring rules, additional Integration Points.
 - b. Refer to the following for guidelines:
 - Oracle RTD platform documentation, which includes *Oracle Real-Time Decisions Getting Started with Oracle RTD*

Your third party implementer documentation, if appropriate

4.1 Aligning Your Business Flow with Oracle RTD

The Base E-Commerce Inline Service includes a collection of integration points that collect real time data about the front-end process (informants) and provide real time recommendations and scores (advisors).

In order to incorporate your own customer data into the Oracle RTD models, it is important to select and order how each applicable integration point can be used in your business process. For example, not all of the provided Advisors (for example, Get Advertisements, Get Up Sell Offers, and so on) may be applicable in the intended workflow. Furthermore, it is critical to identify what data can be passed through the Oracle RTD integration points in real time in the context of the business workflow.

As an early design task, the Inline Service should be evaluated for what pieces are applicable and what data will be available for building the models, either in real time or from data sources. See [Section 4.3, "Mapping Entity Attributes to Customer Data Sources"](#) for further discussions on data mapping.

4.2 Integrating Oracle RTD with the Customer Front End

Several methods for integrating your front end with Oracle RTD are available. These include the use of one of the following:

- Java smart client
- .NET smart client
- Direct web services

Each of these methods is discussed in detail in *Oracle Real-Time Decisions Integration with Oracle RTD*, where you will find steps on completing the integration.

4.3 Mapping Entity Attributes to Customer Data Sources

The Base E-Commerce Inline Service contains a logical entity model that joins common customer and web attributes that a system may gather in the process of navigating through the business workflow.

As a design exercise, customers who use this Inline Service should review the entity structure and map the appropriate attributes to their own data schema. If there are useful data attributes or business objects that are not included in the Base E-Commerce entities, customers should feel free to incorporate them into the entity model.

Data that captures the context of the customer interaction can be extremely useful in understanding customer behavior and offer acceptance. In addition to data captured from a back end data schema, users should also evaluate what information can be captured in real time for the web interaction and be sent to Oracle RTD via the integration points.

Customers can refer to the *Oracle Real-Time Decisions Getting Started Tutorial* for details on adding data sources and modifying or adding entity attributes.

4.4 Adding Additional Logic as Necessary

While the Base E-Commerce Inline Service covers a wide variety of data inputs and touchpoints, it is important to recognize that it serves as a base point. Logic unique to a customer's own business processes may need to be added to basic Oracle RTD operations, such as the following:

- Performing analytics using the Oracle RTD models
- Making recommendations through the Oracle RTD decision engine

The Base E-Commerce Inline Service serves as the foundation where a customer can add custom logic too. Examples of customizations can include the following:

- Additional derived data attributes
- Modified scoring methods for performance goals
- Modification of choice event outcomes
- Additional analytical models to learn on unique business processes
- Additional eligibility or filtering rules for applicable choices

For more information on performing customizations to the Base E-Commerce Inline Service, refer to the following manuals:

- *Oracle Real-Time Decisions Decision Studio Reference Guide*
- *Oracle Real-Time Decisions Getting Started with Oracle RTD*

